Vállalati bérlés- és projektmenedzsment rendszer fejlesztése IT projektmenedzsment szempontok szerint

Székely Dániel

Mérnökinformatikus MSc

2025

Tartalomjegyzék

1. fejezet

Bevezetés

A mai vállalati világban az idő és az erőforrások hatékony kezelése nem csupán optimális, hanem létfontosságú versenyelőny. A hagyományos, papíralapú vagy szigetszerűen kezelt folyamatok lassúak, átláthatatlanok és hibákra hajlamosak. A vállalatoknak ezért exponenciálisan igénye van olyan rendszerekre, amelyek gyorsítják a munkafolyamatokat, csökkentik a hibalehetőségeket és automatizálják a mindennapi működést.

Ebben a környezetben vállaltam egy önálló projektet a vállalatnak, ahol már évek óta tevékenykedem. Egy **egyedi bérléskezelő rendszer (Rental Management System – RMS)** fejlesztését, amely a vállalat minden projektéhez kapcsolódó folyamatait digitális formába önti. A projekt célja nem pusztán a rendszer megvalósítása, hanem annak teljes életciklusának lefedése volt: a tervezéstől és ütemezéstől kezdve a kockázatok feltárásán, a fejlesztésen és a bevezetésen át egészen az üzemeltetésig és karbantartásig.

A dolgozatban ismertetem az RMS céljait, funkcióit és technológiai hátterét. Szakmai szempontból külön hangsúlyt kapnak a projektmenedzsment folyamatok, különösen a **tervezés**, az **ütemezés** és a **kockázatkezelés**, valamint annak bemutatása, hogy ezek miként járultak hozzá a projekt sikeres megvalósításához.

A dolgozat többek között a következő kérdésekre is választ keres:

- Hogyan tervezhető és menedzselhető hatékonyan egy egyedi vállalati szoftverfejlesztési projekt?
- Milyen módszerek és eszközök biztosítják az erőforrások optimális felhasználását és a kockázatok minimalizálását?
- Milyen szakmai tanulságok vonhatók le az önálló projektmenedzsment gyakorlatából, amelyek más projektekben is alkalmazhatók?

A következő fejezetben bemutatom a projekt konkrét célkitűzéseit és főbb jellemzőit, kiemelve, hogy a fejlesztett rendszer milyen problémákat old meg és milyen értéket teremt a vállalat számára.

1.1. ábra. RMS kreatív ábra

2. fejezet

Projekt életciklusa

A projektmenedzsment egyik alapvető elve, hogy minden projekt rendelkezik egy jól definiált életciklussal, amely több, egymásra épülő fázisra bontható. Ez az életciklus nemcsak logikai és időbeli folyamatot ír le, hanem szervezeti, irányítási és ellenőrzési szempontból is meghatározó, hiszen lehetővé teszi a projekt strukturált tervezését, a folyamatok nyomon követését, a kockázatok kezelését és a teljesítmény értékelését [2, 5, 4].

A hazai szakirodalom rendszerint öt alapvető fázist különít el [1, 3]:

1. **Projektindítás (Initiation):** a projekt céljainak, indokoltságának, és a legfontosabb paraméterek meghatározása.
2. **Tervezés (Planning):** részletes ütemezés, erőforrás-tervezés, kockázatelemzés és feladatdefiniálás.
3. **Megvalósítás (Execution):** a projekt tényleges kivitelezése, fejlesztés és implementáció.
4. **Ellenőrzés és irányítás (Monitoring & Controlling):** az előrehaladás, a költségek, az idő és a minőség folyamatos követése.
5. **Lezárás (Closure):** a projekt hivatalos befejezése, átadás, visszacsatolás és tapasztalatok dokumentálása.

2.1. ábra. Projekt kreatív vizualizáció

A fázisok egymásra épülnek, de gyakran átfedésben is zajlanak: például az ellenőrzési és irányítási tevékenység a megvalósítás teljes időtartama alatt folyamatosan jelen van. A modern, iteratív fejlesztési modellek — mint az Agile vagy Scrum — nem feltétlenül követik a klasszikus lineáris struktúrát, hanem sprint-alapú, ciklikus megközelítést alkalmaznak [5, 4].

A klasszikus ötfázisú modell azonban stratégiai és vállalati projekteknél továbbra is nélkülözhetetlen, mivel átlátható keretet biztosít a projekt teljes életciklusára. A **TeDeRMS** fejlesztése során is ez a modell szolgált alapul, amelyet a projekt sajátosságaihoz — önálló fejlesztés, korlátozott erőforrások és vállalati integráció — igazítottam.

2.1. Projektindítás (Initiation)

A projektindítás szakasza a projekt életciklusának alapvető lépése, mivel meghatározza a projekt céljait, irányát és kereteit. A kezdeti fázis során végzett tevékenységek minősége közvetlen hatással van a projekt kockázatkezelésére, a mérföldkövek elérésére és a költséghatékonyságra. A projekt sikeressége nagymértékben függ a kezdeti igényfeltárás alapos elvégzésétől [5, 2, 1, 4].

- **Igényfelmérés:** a vállalat bérléskezelési folyamatait részletesen feltérképeztem, azonosítva a problémás pontokat, és meghatározva a rendszer követelményeit [2].

Az igényfelmérés célja a folyamatok és problémás területek feltárása, amely biztosítja a projekt következő szakaszainak sikeres végrehajtását. A részletes igényfeltárás segít a kockázatok korai azonosításában és a funkciók a valós vállalati igényekhez való illesztésében. Emellett fontos a szervezeti kontextus és az érintettek szempontjainak figyelembevétele, mivel ez növeli a projekt elfogadottságát és a végrehajtás hatékonyságát [1, 4].

- **Célok meghatározása:** világos, mérhető célokat tűztem ki, például az adminisztrációs idő csökkentését, az adatpontosság javítását és az automatizált riportok bevezetését.

A célok meghatározása biztosítja a projekt fókuszát és elősegíti az erőforrások optimális elosztását. A világos, mérhető célok lehetővé teszik a projekt mérföldköveinek nyomon követését és a teljesítmény objektív értékelését. A célok hierarchikus rendszere támogatja a stratégiai, taktikai és operatív döntések összehangolását, valamint a projektmenedzsment folyamatok átláthatóságát [5, 2].

- **Erőforrás-tervezés előkészítése:** meghatároztam a projekt technológiai és időbeli erőforrásigényét. Az önálló fejlesztés miatt kiemelten fontos volt a prioritások és a munkaidő hatékony beosztása.

Az erőforrás-tervezés előkészítése lehetővé teszi a rendelkezésre álló kapacitások optimális elosztását, a munkafolyamatok hatékony szervezését, valamint a prioritások meghatározását. A részletes erőforrás-tervezés segíti a rugalmasság biztosítását, a váratlan események kezelését, és elősegíti a mérföldkövek teljesítését a tervezett idő- és költségkereten belül [1, 4, 5]. A gondos projektindítás tehát stratégiai jelentőségű, mivel biztosítja a projekt későbbi fázisainak sikeres végrehajtását, miközben minimalizálja a félreértésekből és az erőforráspazarlásból eredő kockázatokat [2].

2.2. Tervezés (Planning)

A tervezés szakasza kritikus jelentőségű a projekt sikerének biztosításában, mivel ekkor kerülnek meghatározásra az ütemezés, az erőforrások elosztása és a kockázatok kezelése. A részletes és előrelátó tervezés segít minimalizálni a váratlan problémák hatását, előre jelzi a potenciális kockázatokat, és lehetővé teszi a projekt teljesítményének nyomon követését [5, 2, 1, 4].

- **Ütemterv készítése:** a projekt fő mérföldköveit és feladatait logikai és időbeli sorrendbe állítottam, így a fejlesztés előrehaladása követhetővé vált.

Az ütemterv elkészítése során a feladatok időbeli és logikai összefüggéseinek feltárása lehetővé teszi a projekt átláthatóságát és a folyamatok szisztematikus követését. A jól felépített ütemterv hozzájárul a mérföldkövek teljesítésének ellenőrzéséhez, és segít az erőforrások hatékony allokálásában [1, 4]. Emellett az ütemterv biztosítja a projekt előrehaladásának dokumentálását, amely elengedhetetlen a projektmenedzsment ellenőrzési folyamataihoz.

- **Erőforrás-tervezés:** a napi és technológiai erőforrások meghatározása, a munkabeosztás optimalizálása biztosította az önálló fejlesztés gördülékenységét.

Az erőforrás-tervezés során a rendelkezésre álló humán és technológiai kapacitások felmérése és optimális elosztása kulcsfontosságú. A részletes erőforrás-tervezés lehetővé teszi a projekt rugalmasságát, elősegíti a prioritások meghatározását, és csökkenti a késedelmek kockázatát [5, 2]. Különösen kiscsapatos vagy önálló fejlesztési környezetben a munkabeosztás és a rendelkezésre álló idő hatékony kezelése elengedhetetlen a projekt gördülékeny megvalósításához.

2.2. ábra. Naptári ütemterv vizualizáció

- **Kockázatelemzés:** azonosítottam a fő kockázatokat (technikai hibák, adatvesztés, üzleti logika hibák), és kidolgoztam a megelőző és elhárító intézkedéseket.

A kockázatelemzés során feltárt potenciális problémák előrejelzése és kezelési tervének kidolgozása elengedhetetlen a projekt biztonságos és hatékony végrehajtásához. A kockázatmátrix alkalmazása lehetővé teszi a valószínűség és a hatás értékelését, és segíti a megelőző intézkedések priorizálását [1, 4]. Ez a megközelítés hozzájárul a projekt stabilitásához, és minimalizálja a váratlan események negatív következményeit. A tervezési szakasz során alkalmazott módszerek, mint a feladatlista, az ütemezett mérföldkövek és a kockázatmátrix, valamint a folyamatos önellenőrzés és visszacsatolás, biztosítják a projekt strukturált és ellenőrizhető előrehaladását [5, 2].

2.3. Megvalósítás (Execution)

A megvalósítás szakasza a projekt tényleges kivitelezését foglalja magában, amely során a tervezett funkciók, folyamatok és erőforrások gyakorlati alkalmazásra kerülnek. Ebben a fázisban a projektmenedzsment és a technikai végrehajtás szoros összhangja kulcsfontosságú a kívánt eredmények eléréséhez [5, 2, 4].

- **Fejlesztési folyamatok:** moduláris architektúra kialakítása, backend és frontend párhuzamos fejlesztése, verziókövetés GitHub-on.

A moduláris architektúra lehetővé teszi a fejlesztési folyamatok párhuzamosítását, a hibák izolálását és a könnyebb karbantartást. A moduláris felépítés, valamint a verziókövető rendszerek alkalmazása elősegíti a projekt kontrollját, biztosítja a kód integritását és csökkenti a fejlesztési hibák számát [1, 4]. A backend és frontend párhuzamos fejlesztése hatékonyabb erőforrás-kezelést és gyorsabb fejlesztési ütemet tesz lehetővé.

- **Tesztelés:** folyamatos unit és funkcionális tesztek, a rendszer stabilitásának és megbízhatóságának biztosítására.

A tesztelés kiemelt jelentőségű a megvalósítás során, mivel biztosítja a rendszer funkcionalitását, stabilitását és a hibamentes működést. A folyamatos tesztelés csökkenti a hibák későbbi előfordulásának kockázatát, és elősegíti a fejlesztés iteratív javítását, valamint a minőségbiztosítási folyamatok integrációját [2, 5]. A unit tesztek elsősorban az egyes modulok helyes működését ellenőrzik, míg a funkcionális tesztek a teljes rendszer üzleti logikáját vizsgálják.

- **Dokumentáció:** részletes kód- és rendszer-dokumentáció, amely elősegíti a későbbi karbantartást és bővítést.

A dokumentáció a projekt hosszú távú fenntarthatóságának alapfeltétele. A kód és rendszer részletes dokumentálása nemcsak a karbantartást és bővítést könnyíti meg, hanem elősegíti a tudás megőrzését, a fejlesztői csapat tagjai közötti kommunikációt, és támogatja az esetleges auditok vagy rendszerellenőrzések lebonyolítását [1, 4, 2]. A megvalósítás szakasz tehát biztosítja, hogy a rendszer minden funkciója a tervek szerint valósuljon meg, és a vállalat igényeit teljes mértékben kielégítse, miközben a projektmenedzsment eszközök és a dokumentáció révén a folyamat átlátható és ellenőrizhető marad [5].

2.4. Ellenőrzés és irányítás (Monitoring és Controlling)

Az ellenőrzés és irányítás fázisa a projekt sikerének egyik legfontosabb garanciája, mivel biztosítja, hogy a tervezett célok és mérföldkövek teljesüljenek, az erőforrások optimálisan legyenek felhasználva, és a projekt kockázatai időben kezelhetők legyenek [5, 2, 1, 4]. A monitoring és controlling nemcsak a problémák azonosítását szolgálja, hanem lehetővé teszi a proaktív beavatkozást és a projekt folyamatos optimalizálását.

- **Haladás nyomon követése:** mérföldkövek teljesülésének ellenőrzése, eltérések azonosítása és korrekciója.

A haladás nyomon követése során a mérföldkövek teljesítését folyamatosan ellenőrizni kell, hogy a projekt a tervezett ütemterv szerint haladjon. A rendszeres ellenőrzés lehetővé teszi a korai eltérések azonosítását, a késedelmek és a túlzott erőforrásfelhasználás megelőzését, valamint támogatja a szükséges korrekciók gyors végrehajtását [1, 4]. A mérföldkövek objektív értékelése javítja a projekt átláthatóságát és a vezetői döntések megalapozottságát.

- **Kockázatok kezelése:** a kockázatmátrix folyamatos frissítése, a problémák gyors és hatékony megoldása.

A kockázatok folyamatos kezelése lehetővé teszi, hogy a potenciális problémák hatása minimalizálható legyen. A kockázatmátrix rendszeres frissítése segít az új kockázatok

azonosításában, a valószínűség és hatás értékelésében, valamint a megelőző intézkedések és vészforgatókönyvek kidolgozásában [5, 2]. A proaktív kockázatkezelés csökkenti a projekt idő- és költségkeretének túllépésének esélyét, és növeli a projekt sikerességét.

2.3. ábra. Kockázatkezelés példa

- **Minőségellenőrzés:** a rendszer funkcionalitásának és stabilitásának folyamatos ellenőrzése a hibák minimalizálása érdekében.

A minőségellenőrzés célja, hogy a projekt kimenete megfeleljen a tervezett specifikációknak, és a hibák a lehető legkorábban kerüljenek azonosításra. A folyamatos minőségellenőrzés biztosítja a rendszer stabilitását és megbízhatóságát, támogatja az iteratív javításokat, és hozzájárul a projekt végső sikeréhez [1, 4, 2]. Az ellenőrzés és irányítás tehát kulcsfontosságú a projekt kereteinek betartásában, a mérföldkövek teljesítésében és a kívánt minőség biztosításában, miközben lehetőséget nyújt a folyamatos optimalizálásra és a kockázatok proaktív kezelésére.

2.5. Projekt lezárás (Closure)

A projekt lezárása a projektmenedzsment egyik kritikus fázisa, mivel ekkor valósul meg a fejlesztett rendszer végső átadása, a tapasztalatok összegzése és a hosszú távú fenntarthatóság biztosítása. A lezárás során történő alapos értékelés és dokumentáció kulcsfontosságú a projekt teljesítményének, hatékonyságának és minőségének fenntartásához [5, 2, 1, 4].

- **Rendszer átadása:** a **TeDeRMS** teljes körű telepítése a vállalat környezetében.

A rendszer átadása során a telepítésnek zökkenőmentesen kell biztosítania a teljes funkcionalitást a vállalati környezetben. Az átadás minősége és pontossága közvetlen hatással van a felhasználói elfogadásra, a napi operatív folyamatokra és a projekt hosszú

távú sikerére [2, 1]. A hibamentes átadás elősegíti a vállalati folyamatok zavartalan működését, és csökkenti a támogatási igényeket.

- **Dokumentáció:** részletes felhasználói és adminisztrátori kézikönyvek készítése.

A dokumentáció a hosszú távú fenntarthatóság záloga, mivel biztosítja, hogy a rendszer működését és konfigurációját később bárki könnyen megérthesse és kezelhesse. A dokumentáció nemcsak a felhasználók számára nyújt útmutatót, hanem a karbantartást és a jövőbeli fejlesztéseket is támogatja, valamint alapot ad a rendszer auditálásához [5, 4].

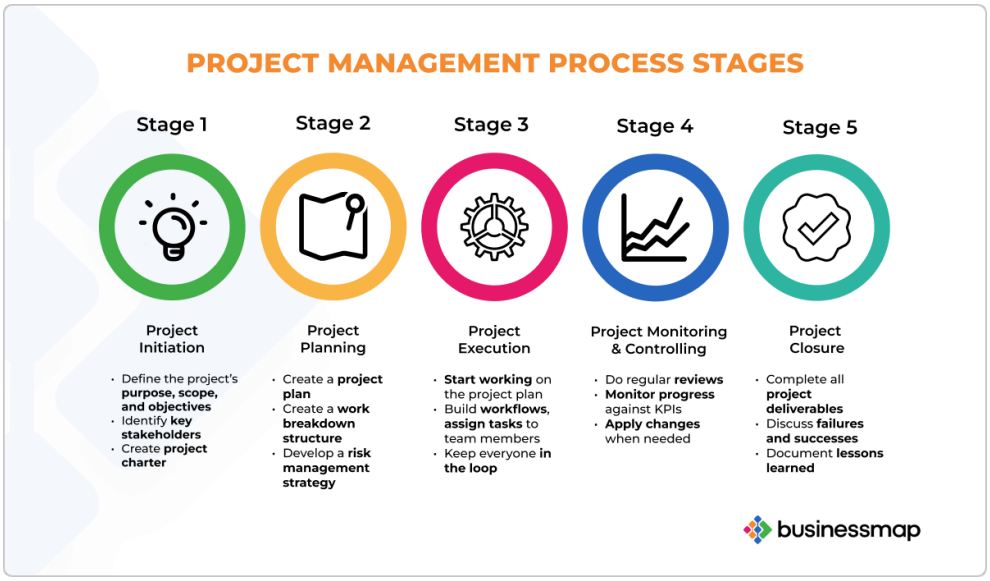
- **Oktatás:** a kulcsfelhasználók képzése a rendszer hatékony használatára.

A képzés biztosítja, hogy a rendszer a vállalatnál maximális hatékonysággal legyen alkalmazható. A felhasználói oktatás közvetlenül hozzájárul a rendszer elfogadásához, a hibák csökkentéséhez és a napi működés zavartalanságához [2, 1]. Az oktatás során szerzett ismeretek elősegítik a rendszer önálló kezelését és a belső támogatási igény minimalizálását.

- **Tanulságok összegzése:** a projekt során szerzett tapasztalatok dokumentálása és javaslatok a jövőbeli fejlesztésekhez.

A projekt lezárása során a tanulságok összegzése lehetővé teszi a szervezeti tudás megőrzését és a folyamatok fejlesztését. A tapasztalatok dokumentálása segíti a jövőbeli projektek tervezését, a hibák elkerülését és a hatékonyabb projektmenedzsment kialakítását [5, 2, 4]. Ez a tudás hosszú távon növeli a vállalati projektek sikerességét és stabilitását.

A lezárási fázis tehát biztosítja, hogy a rendszer hosszú távon stabilan és hatékonyan működjön, miközben a projektmenedzsment tapasztalatai a későbbiekben is hasznosíthatók.



2.4. ábra. Projektmenedzsment folyamatok fázisai

3. fejezet

Projekt áttekintése és részletes bemutatása

3.1. A vállalat és az üzleti igények

A projekt célja a **TéDé Rendezvények** vállalat bérlés és projektkezelési folyamatainak teljes digitalizálása volt. A korábbi papíralapú és Excel-alapú folyamatok nem feleltek meg a vállalat növekvő igényeinek, és egy modern, integrált rendszer kialakítása vált szükségessé. Az online elérhető üzenetküldő és felhő alapú megoldások nem kínáltak megfelelő rugalmasságot és testreszabhatóságot, ezért egy egyedi fejlesztésű rendszer mellett született döntés.

A vállalat számára kiemelten fontos volt egy egységes, modern és felhasználóbarát rendszer, amely:

- automatizálja a bérlési folyamatokat,
- nyomon követi a készleteket,
- biztosítja az adminisztráció teljes körű kezelését,
- egyszerűsíti a munkafolyamatokat,
- csökkenti a hibalehetőségeket,
- javítja a kommunikációt a csapaton belül,
- egységesíti a dokumentációt,
- minden információ egy helyen elérhető,
- lehetővé teszi a gyors árajánlatkészítést.

További célok voltak:

- modern, felhasználóbarát felület
- egyszerűen telepíthető és skálázható a **Docker** segítségével
- könnyen karbantartható és bővíthető architektúra
- biztonságos hozzáférés-kezelés és jogosultságok
- részletes riportálási és statisztikai funkciók

3.2. A projekt összetettsége és kihívásai

A projekt felettébb összetettnek tekinthető, mivel:

- többféle felhasználói szerepkört kellett kezelnie (admin, munkatárs, menedzser),
- integrálni kellett különböző adatforrásokat és bérlési folyamatokat,
- biztonsági és hozzáférés-kezelési követelményeknek kellett megfelelnie,
- a fejlesztés során több technológiát kellett összehangolni a rugalmasság és megbízhatóság érdekében.

3.3. Technológiai háttér

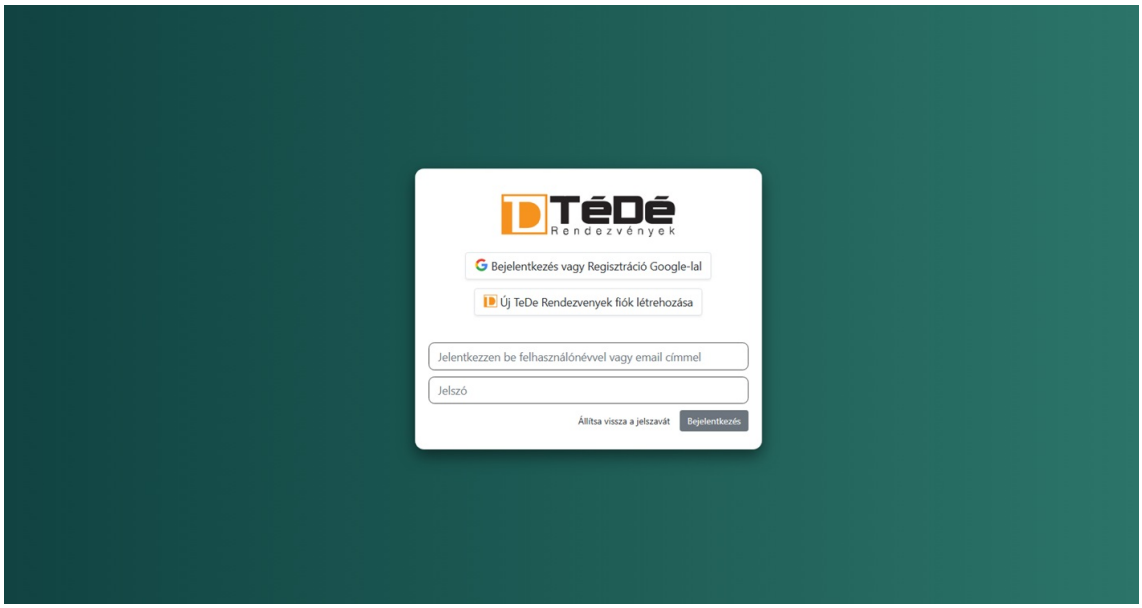
A rendszer fejlesztése a következő technológiákra épült:

- **Backend:** PHP a Twig sablonmotorral,
- **Frontend:** modern responsive felhasználói felület Twig sablonokkal,
- **Konténerizálás és telepítés:** Docker és Docker Compose, előre elkészített konténerekben,
- **Konfiguráció:** testreszabható `.env` fájlok segítségével.
- **Reverse proxy:** Nginx a kérések kezelésére és a statikus fájlok kiszolgálására,
- **Adatbázis:** MySQL a megbízható adatkezelés érdekében,
- **Verziókezelés:** Git a kód nyomon követésére.

3.4. A megvalósított szoftver

3.4.1. Bejelentkezés

A Bejelentkezési felület lehetővé teszi a felhasználók számára, hogy biztonságosan hozzáférjenek a rendszerhez. Ezt megtehetik helyi fiókkal vagy Google OAuth2 hitelesítéssel, amely egyszerűsíti a bejelentkezési folyamatot és növeli a biztonságot. A bejelentkezés után amennyiben új felhasználó lép be a rendszerbe, egy egyszeri csatlakozó jelszó megadására van szükség amivel kapcsolódik a vállalathoz, ezzel biztosítva, hogy csak jogosult személyek férjenek hozzá a rendszerhez, és minden tevékenység naplózásra kerül a későbbi ellenőrzés érdekében.



3.1. ábra. Bejelentkezés

3.4.2. Dashboard

Az alkalmazás indítása után az alapértelmezett nézet a **Dashboard**, amely a rendszer központi irányítópultjaként szolgál. Célja, hogy a felhasználó egyetlen felületen, átlátható formában kapjon átfogó képet a vállalkozás aktuális működéséről, az eszközállományról, a projektek státuszáról és a saját napi tevékenységeiről.

A felület moduláris felépítésű, azaz minden információ önálló panelen jelenik meg, így a felhasználó gyorsan hozzáférhet az őt érdeklő adatokhoz anélkül, hogy menük között kellene navigálnia. A legfontosabb panelek a következők:

- **Eszköz statisztika:** összesített információkat jelenít meg az aktuális eszközpark állapotáról, beleértve az összértéket, az össztömeget, a raktárban lévő mennyiséget

és az eszköztípusok számát. Ez különösen hasznos a logisztikai és bérléskezelési döntések előkészítésénél.

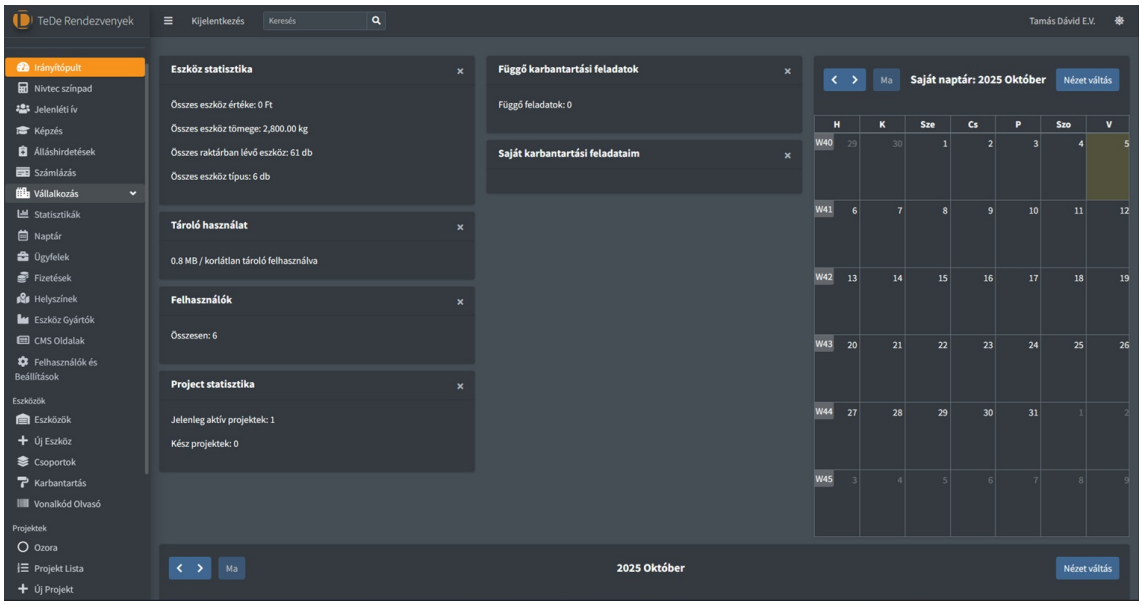
- **Tároló használat:** kijelzi a tárhely aktuális kihasználtságát, ami segíti a rendszeradminisztrátort a tárolókapacitás hatékony felügyeletében, különösen a feltöltött képek és dokumentumok kezelésénél.
- **Felhasználók:** megjeleníti az aktív felhasználók számát, ezzel is elősegítve a jogosultságkezelést és a hozzáférések ellenőrzését.
- **Projekt statisztika:** az éppen aktív és a lezárt projektek számát mutatja, ami egyértelmű képet ad a vállalat aktuális munkaterheléséről és projektfázisairól.
- **Karbantartási feladatok:** listázza a függő karbantartási műveleteket, illetve a bejelentkezett felhasználó saját feladatait, ezzel biztosítva, hogy a rendszerüzemeltetés és az eszközkarbantartás ne maradjon el.

A jobb oldalon található **naptárnézet** a Dashboard egyik kulcsfunkciója, amely a tervezés és szervezés hatékonyságát támogatja. A naptár kétféle módon használható:

- **Vállalati nézet:** az összes eseményt, projektmérföldkövet és karbantartási ütemezést megjeleníti, így a vezetők és adminisztrátorok teljes képet kapnak a szervezet aktuális feladatairól.
- **Személyes nézet:** kizárólag az adott felhasználóhoz rendelt eseményeket és határidőket mutatja, ezzel segítve az egyéni időbeosztás és feladatprioritás kezelését.

A naptár interaktív: a felhasználók közvetlenül a felületen lépkedhetnek a hetek és hónapok között, illetve az eseményekhez kapcsolódó részletek is megtekinthetőek. Ez a funkció jelentősen növeli a munkaszervezés hatékonyságát, mivel vizuális formában támogatja a tervezést és a határidők követését.

A Dashboard tehát nem csupán egy információs felület, hanem a rendszer működésének központi irányítási pontja, amely valós időben tükrözi a vállalkozás operatív állapotát. Felépítése reszponzív, így asztali és mobil eszközön egyaránt optimálisan használható, miközben a modulok dinamikusan frissülnek a háttérben futó adatbázis-lekérdezések alapján.



3.2. ábra. Dashboard – vállalati áttekintő felület

3.4.3. Eszközök kezelése

Az **Eszközök** modul a rendszer egyik legfontosabb eleme, hiszen itt történik a teljes készlet nyilvántartása, kezelése és karbantartása. A felület célja, hogy a vállalat munkatársai gyorsan és átlátható módon kezelhessék a több száz, esetenként több ezer eszközből álló technikai parkot, valós idejű szűrési és rendezési lehetőségek mellett.

A rendszerben minden eszköz önálló entitásként szerepel, amelyhez részletes metaadatok és kapcsolódó információk tartoznak. Az eszközök az alábbi attribútumokkal rendelkeznek:

- **Név és kategória:** az eszköz megnevezése és típusbesorolása (pl. hangtechnika, fénytechnika, videótechnika).
- **Címke és egyedi azonosító:** egyedi vonalkódos címkével vagy belső ID-vel azonosítható minden eszköz, ami megkönnyíti a gyors leltározást és az eszközmozgások követését.
- **Gyártó és modell:** a pontos technikai paraméterezés és kompatibilitás érdekében.
- **Sorozatszám és állapot:** minden eszköz karbantartási vagy garanciális nyilvántartásának alapja.
- **Leírás és karbantartási információk:** részletes technikai leírás, valamint az eszköz állapotának és karbantartási ciklusainak dokumentálása.
- **Mennyiség és bérlési árak:** az aktuális készlet és a bérlésre vonatkozó díjszabás nyilvántartása.

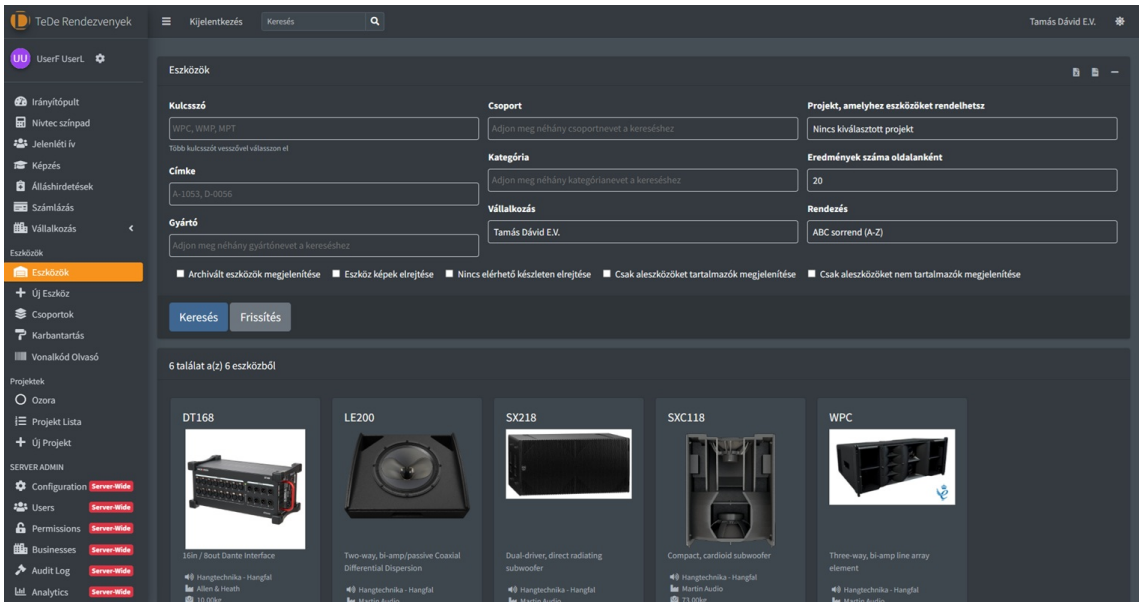
- **Kép:** vizuális azonosítást segítő fotó, amely a felhasználói élményt és a hibamentes választást támogatja.

A felület fejlett szűrési és keresési funkciókat biztosít, így a felhasználók pillanatok alatt megtalálhatják a keresett eszközt akár több száz tétel között is. A szűrési lehetőségek a következők:

- **Kulcsszó alapú keresés:** akár részleges egyezés alapján is szűr a név, címke vagy gyártó szerint.
- **Csoport és kategória szerinti szűrés:** lehetővé teszi, hogy csak adott projekttípushoz vagy technikai kategóriához tartozó eszközök jelenjenek meg.
- **Projekt szerinti rendezés:** megjeleníthetők kizárólag egy adott eseményhez, produkcióhoz vagy ügyfélhez rendelt eszközök.
- **Vállalkozás szűrő:** a különböző alvállalkozókhoz tartozó eszközök elkülönített kezelése.
- **Előre beállított nézetek:** lehetőség van archivált eszközök megjelenítésére, képek elrejtésére vagy a készleten nem elérhető tételek kiszűrésére.
- **Rendezési opciók:** ABC sorrend, gyártó, kategória vagy elérhetőség alapján történő megjelenítés.
- **Speciális szűrők:** csak aleszközöket (pl. kábelek, tartozékok) vagy csak főeszközöket tartalmazó lista megjelenítése.

A keresési rendszer valós időben frissül, így a felhasználó azonnal látja az eredményeket, miközben a rendszer automatikusan optimalizálja a lekérdezéseket a szerver teljesítményének megőrzése érdekében. Az eszközlista kártyás formában jelenik meg, vizuálisan elkülönítve a különböző típusokat és státuszokat, így a kezelő személyzet gyorsan áttekintheti a rendelkezésre álló készletet.

Az **Eszközök** modul tehát nem csupán egy adatbázis-kezelő felület, hanem egy interaktív, rugalmas eszközmenedzsment-rendszer, amely az operatív döntéshozatalt és a napi logisztikai munkát egyaránt támogatja.



3.3. ábra. Eszközök kezelése

3.4.4. Projektek kezelése

A rendszer egyik legfontosabb modulja a **Projektek kezelése**, amely a vállalkozás teljes operatív működését átfogja. Ebben a nézetben a felhasználó részletesen konfigurálhatja és nyomon követheti az egyes eseményekhez, rendezvényekhez vagy megrendelésekhez tartozó projekteket. A cél, hogy minden információ, az ügyféltől kezdve az eszközökön és helyszíneken át egészen a csapatbeosztásig, egy központi felületen legyen kezelhető.

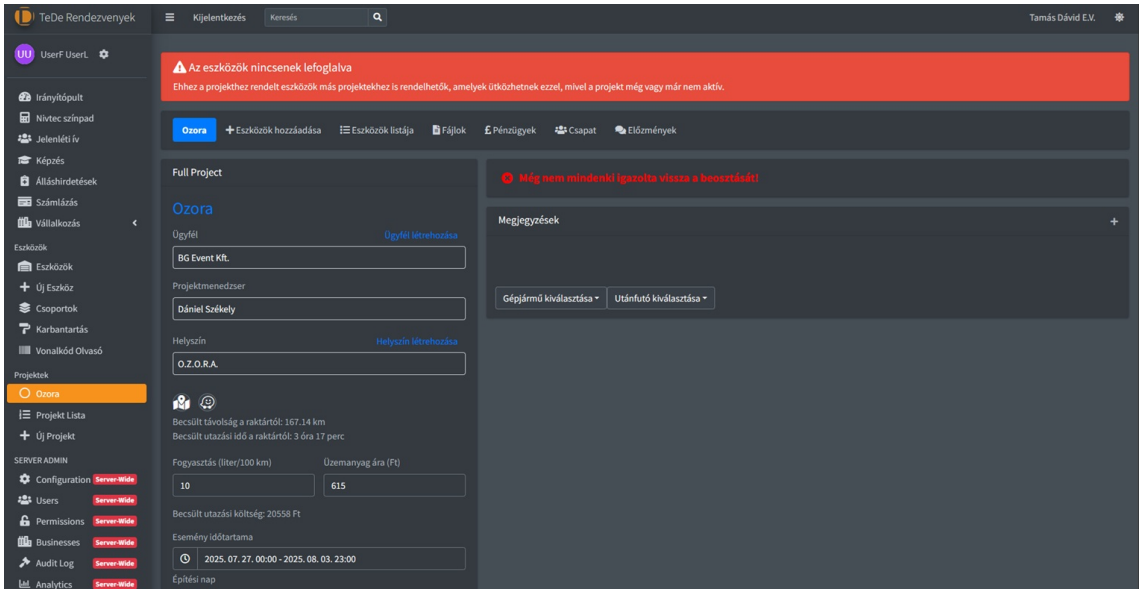
A projekt adatlapja több logikai szekcióra tagolódik:

- **Ügyfélkezelés:** a projekthez tartozó ügyféladatokat jeleníti meg, illetve új ügyfél is létrehozható közvetlenül innen. Ez leegyszerűsíti a megrendelések adminisztrációját és a CRM-folyamatok integrációját.
- **Projektmenedzser és felelősök:** minden projekthez hozzárendelhető egy vagy több felelős személy, akik a feladatok koordinálásáért és a végrehajtás felügyeletéért felelősek. A rendszer automatikus értesítéseket küld, ha egy új projektet hoznak létre vagy módosítanak.
- **Helyszínkezelés:** a rendezvény vagy telepítés pontos helyszínét tartalmazza. Az adatbázis képes korábbi helyszínek újrafelhasználására, így a gyakran ismétlődő helyszínek (pl. fesztiválok, partnerrendezvények) gyorsan kiválaszthatók.
- **Logisztikai kalkuláció:** a rendszer automatikusan kiszámítja a raktártól való távolságot, az utazási időt, az üzemanyag-felhasználást és a várható utazási költséget. Ez

a funkció különösen fontos a terepi projektek (például rendezvényhelyszínek) előkészítése során, mivel a logisztikai tervezés a költségvetés egyik kritikusabb eleme.

- **Erőforrás-hozzárendelés:** a projektfelülethez közvetlenül csatolhatók eszközök, járművek és utánfutók. A rendszer automatikusan figyelmeztet, ha egy eszköz már másik projekthez van rendelve, vagy ha a projekt inaktív, így elkerülhetők az ütközések és a duplikációk.

- **Eseményidőtartam és ütemezés:** a projekt időtartamát kezdő- és záródátum alapján rögzíti a rendszer, amely később összekapcsolódik a naptármodullal. A felhasználók így a teljes munkafolyamatot, az előkészítéstől az építési napig, kronológiai sorrendben követhetik.



3.4. ábra. Projekt részletes adatlap amit az adminok és menedzserek látnak

A projektekhez további funkciók is társulnak, például fájlkezelés (dokumentumok, szerződések, műszaki rajzok feltöltése), pénzügyi modul (költségvetés és elszámolás), valamint **csapatkezelés**, amelyben a rendszer minden résztvevőt listáz és státuszukat (jóváhagyott / függő) megjeleníti. Ezáltal a projektmenedzser azonnal látja, ha valaki még nem erősítette meg a beosztását, ezt vizuálisan is jelzi a felület figyelmeztető sávja.

A **Projektek kezelése** modul így nem csupán adminisztrációs célokat szolgál, hanem valódi menedzsmenteszközzé válik, amely integrálja a logisztikai, humán és technikai folyamatokat. Ezzel biztosítja, hogy minden esemény előkészítése és lebonyolítása transzparensen, ellenőrzötten és hatékonyan történjen.

A rendszer minden egyes módosítást naplóz, így a projekt életciklusa teljes egészében visszakövethető, ki mikor és milyen változtatást hajtott végre. Ez különösen fontos a

későbbi auditok és elemzések szempontjából, valamint amennyiben vitás helyzetek merülnének fel a projekt végrehajtása során, hiszen minden lépés dokumentált és ellenőrizhető marad.

3.5. ábra. Projekt kompakt adatlap amit a munkavállalók látnak

3.4.5. GitHub és Reverse Proxy

A verziókezelés és a biztonságos hálózati hozzáférés is kiemelt szerepet kapott a projekt során. A projekt forráskódja a **GitHub** platformon található, amely lehetővé teszi a verziók követését, a biztonsági mentések készítését és a folyamatos fejlesztést. A GitHub használata növeli a fejlesztési folyamat átláthatóságát, támogatja a csapatmunkát, és biztosítja a kód minőségének megőrzését.

A fejlesztés egy főágon (`main branch`) zajlik, ehhez a lokális környezetből `git push/pull` műveletekkel történik a szinkronizálás. Minden új funkció fejlesztése külön fejlesztői ágon (`feature branch`) történik, ezáltal a stabil főverzióba csak tesztelt és hibamentes kód kerülhet. A commit-ok egységes elnevezési sémát követnek (pl. `feat:`, `fix:`, `refactor:`), ami megkönnyíti a változások visszakövetését és a hibakeresést. A verziókezelés így nem csupán biztonsági, hanem projektmenedzsment-eszközként is működik.

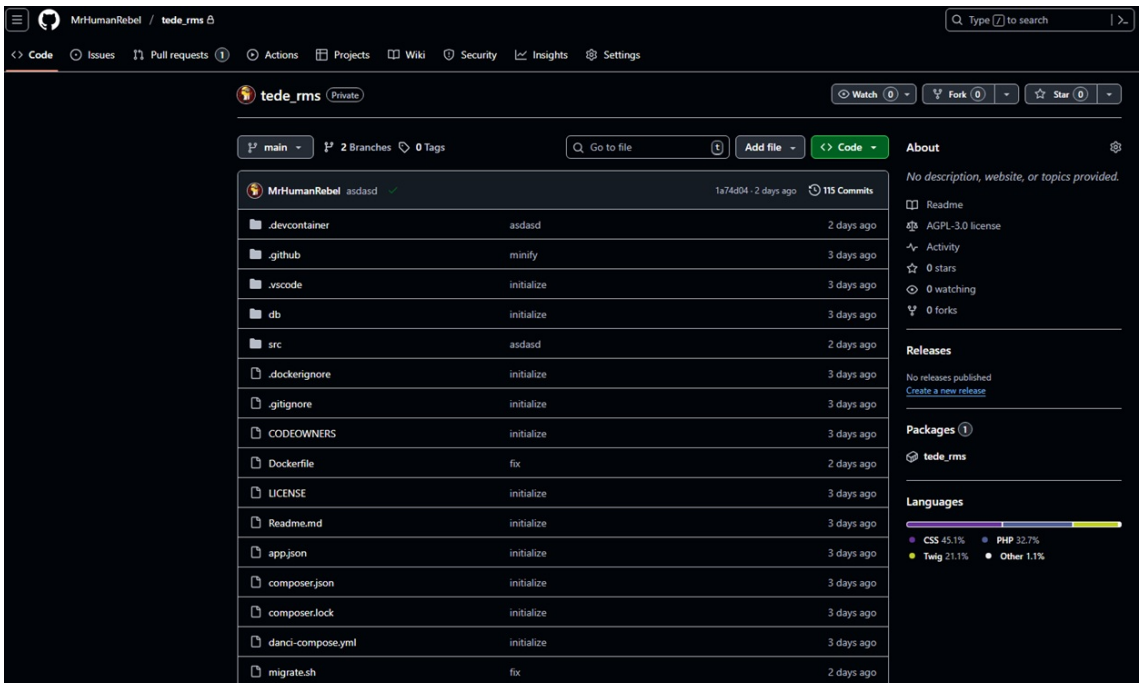
A fejlesztési környezet a **Docker Compose** infrastruktúrán alapul, amely biztosítja a konténerek gyors indítását, újratelepítését és frissítését. A rendszer több szolgáltatásból áll — például webalkalmazás, adatbázis és proxy réteg — amelyek egymással biztonságosan kommunikálnak a konténerhálón belül. A külső hálózat felé a kapcsolatot egy **Reverse Proxy** réteg biztosítja, amely a beérkező HTTP(S) kéréseket a megfelelő szolgáltatáshoz irányítja.

A reverse proxy réteg **NGINX** alapokon működik, amely:

- biztosítja a HTTPS titkosítást és a tanúsítványkezelést (`Let's Encrypt`);
- lehetővé teszi az aldomének és portok dinamikus irányítását;
- csökkenti a külső támadások kockázatát az alkalmazásrétegek elszigetelésével;
- elősegíti a terheléselosztást, ezáltal növeli a rendszer válaszképességét és stabilitását.

A proxy réteg nemcsak hálózati védelmet nyújt, hanem támogatja a fejlesztés és üzemeltetés szétválasztását is. A GitHub repository tartalmazza a `docker-compose.yml` és `nginx.conf` fájlokat, amelyek lehetővé teszik a rendszer gyors telepítését bármely környezetben — fejlesztői, teszt vagy éles szerveren — egyetlen parancs segítségével.

Ez az architektúra a modern **DevOps-elvek** alapján működik, ahol a fejlesztés, verziókezelés, biztonság és üzemeltetés integrált egységet alkot. A GitHub–Docker–NGINX kombináció garantálja a kód stabilitását, az infrastruktúra rugalmasságát és az adatbiztonságot, ami alapvető követelmény egy vállalati szintű rendszer esetében.



3.6. ábra. GitHub repository

4. fejezet

Kockázatkezelés és problémamegoldás

A **TeDeRMS** projekt során a kockázatkezelés kulcsfontosságú szerepet játszott, mivel a projekt önálló fejlesztésként valósult meg, így minden döntés és probléma gyors és hatékony kezelést igényelt. A kockázatkezelés célja a potenciális problémák előrejelzése, azok hatásának minimalizálása, valamint a projekt sikerének biztosítása volt.

4.1. Felmerült problémák

A fejlesztés során számos kihívás és nehézség merült fel, amelyek kezelése kulcsfontosságú volt a projekt sikeréhez és a rendszer funkcionalitásának biztosításához. A tapasztalatok alapján megfigyelhető, hogy a projektmenedzsment klasszikus fázisai jól előre jelzik a problémák jellegét, de a gyakorlatban gyakran szükség van rugalmasságra és kreatív megoldásokra is [5, 2].

- **Technológiai problémák:** a rendszer stabilitása és a Docker-konténerek kezdeti konfigurációja nem volt optimális, ami futtatási hibákhoz vezetett.

Ez a probléma rámutatott arra, hogy a modern szoftverfejlesztési környezetekben a technológiai infrastruktúra optimalizálása legalább olyan fontos, mint a szoftver logikájának megtervezése. A magyar szakirodalom hangsúlyozza, hogy a stabil technológiai alap kritikus a rendszer megbízhatósága szempontjából, és a kezdeti hibák megfelelő dokumentációval és iteratív konfigurálással kezelhetők [1, 4]. Saját tapasztalatom szerint az ilyen problémák korai felismerése jelentősen csökkentette a későbbi üzemeltetési nehézségeket.

- **Funkcionális kihívások:** a bérléskezelési folyamatok pontos leképezése a szoftverben nehézkes volt, különösen a különböző státuszok és jogosultságok kezelése.

A funkcionalitás pontos implementálása során szembesültem azzal, hogy a vállalati folyamatok összetettsége gyakran túlmutat a papíron megadott szabályokon. A szakiro-

dalom szerint a rendszertervezés során a folyamatok részletes feltérképezése és a valós működés elemzése elengedhetetlen a hibák minimalizálásához [2, 5]. Saját értékelésem szerint a moduláris architektúra bevezetése és a folyamatok iteratív tesztelése kulcsfontosságú volt a pontos implementáció eléréséhez.

- **Időmenedzsment:** a projekt önálló végzése miatt a napi munka és a projekt előrehaladása közötti egyensúly fenntartása komoly kihívást jelentett.

Az időmenedzsment kérdése kiemelten fontos volt, mivel a korlátozott erőforrások és a napi feladatok összehangolása folyamatos priorizálást igényelt. A hazai szakirodalom rámutat, hogy a projekt időbeli koordinációja, a mérföldkövek betartása és a személyes produktivitás optimalizálása kritikus tényezők az önálló fejlesztések során [4, 1]. Saját tapasztalatom alapján a heti tervezés és a feladatok rendszeres felülvizsgálata segített a haladás fenntartásában.

- **Tesztelés és hibajavítás:** a rendszer moduláris felépítése miatt a hibák lokalizálása és javítása több iterációt igényelt.

A hibák azonosítása és javítása a moduláris felépítés miatt komplex feladat volt, mivel egy-egy modul problémája láncreakcióként hatott a rendszer más részeire. A szakirodalom kiemeli, hogy a folyamatos unit- és integrációs tesztelés, valamint a hibák dokumentálása és visszacsatolása alapvető a projekt stabilitása szempontjából [5, 2]. Saját megfigyelésem szerint az iteratív hibajavítás és a részletes tesztelési protokollok alkalmazása lehetővé tette a moduláris rendszer hatékony működését, miközben a tanult módszerek későbbi projektekben is alkalmazhatók.

Összességében a felmerült problémák rávilágítottak arra, hogy az önálló fejlesztés során a technológiai, funkcionális és menedzsment kihívások integrált kezelése kulcsfontosságú, és a magyar szakirodalom által javasolt módszertani eszközök alkalmazása jelentősen növelte a projekt sikerességét és a rendszer minőségét.

4.2. Kockázatok azonosítása és priorizálása

A projekt kezdeti fázisában kiemelten fontos volt a potenciális problémák előrejelzése és kezelése, mivel a kockázatok megfelelő azonosítása és priorizálása közvetlen hatással van a projekt sikerére és a rendszer minőségére. A szakirodalom szerint a kockázatmenedzsment alapja a kockázatok strukturált feltérképezése, az értékelésük és a megfelelő megelőző intézkedések kialakítása [5, 2, 1].

A projekt során egy **kockázatmátrix** került kialakításra, amely azonosította a legkritikusabb problémákat és segítette a megfelelő prioritások meghatározását:

- **Magas prioritású kockázatok:** kritikus hibák a backend működésében, adatvesztés, biztonsági hiányosságok.

Ezek a kockázatok a rendszer alapvető működését fenyegették, ezért azonnali figyelmet igényeltek. A magyar szakirodalom hangsúlyozza, hogy a magas prioritású problémák kezelése kulcsfontosságú a rendszer stabilitásának és megbízhatóságának biztosításához [4, 2]. Saját tapasztalatom alapján a kritikus hibák korai azonosítása és a preventív intézkedések bevezetése jelentősen csökkentette a rendszer működési kockázatát.

- **Közepes prioritású kockázatok:** kisebb felhasználói felületbeli problémák, riportálási hibák.

A közepes prioritású kockázatok a felhasználói élményt és a riportok pontosságát érintették. A szakirodalom kiemeli, hogy ezek a problémák bár nem veszélyeztetik közvetlenül a rendszer működését, hosszú távon befolyásolhatják az elfogadást és a felhasználói elégedettséget [5, 1]. Saját értékelésem szerint ezeknek a hibáknak a folyamatos monitorozása és iteratív javítása lehetővé tette a felhasználói élmény optimalizálását anélkül, hogy a kritikus funkciókat veszélyeztettük volna.

- **Alacsony prioritású kockázatok:** kisebb esztétikai hibák vagy későbbi bővítési igények.

Az alacsony prioritású kockázatok elsősorban a rendszer megjelenését vagy a jövőbeli fejlesztések ütemezését érintették. A szakirodalom hangsúlyozza, hogy ezek a kockázatok ugyan kevésbé sürgetők, de hosszú távon hozzájárulnak a rendszer karbantarthatóságához és a fejlesztési terv fenntarthatóságához [2, 4]. Saját tapasztalatom alapján az alacsony prioritású problémák nyomon követése lehetővé tette, hogy a projekt fókusza a kritikus funkciók biztosításán maradjon, miközben a jövőbeli bővítések előkészítése is folyamatban volt.

Összességében a kockázatok azonosítása és priorizálása lehetővé tette, hogy a projekt során először a legkritikusabb problémákra koncentráljak, ezáltal a rendszer stabilitása és funkcionalitása biztosított volt. A strukturált kockázatmenedzsment alkalmazása és a folyamatos monitorozás a magyar szakirodalom szerint elengedhetetlen a sikeres projektvégrehajtáshoz [5, 1, 2, 4].

4.3. Megoldási stratégiák

A felmerült problémák kezelése során a cél a rendszer stabilitásának, funkcionalitásának és a projekt határidőinek biztosítása volt. A szakirodalom hangsúlyozza, hogy a

problémákra alkalmazott stratégiák strukturáltsága és következetessége jelentősen befolyásolja a projekt sikerét [5, 2, 4].

A technológiai problémák, különösen a Docker-konténerek és a PHP/Twig környezet konfigurációja kezdetben instabilitást okoztak, ami futtatási hibákhoz vezetett. A szakirodalom szerint az iteratív finomhangolás és a rendszeres tesztelés kulcsfontosságú a szoftver stabilitásának biztosításához [1]. Saját tapasztalatom alapján a konfigurációk fokozatos javítása, a hibák dokumentálása és a rendszeres tesztkörök alkalmazása jelentősen csökkentette a technikai kockázatot.

A funkcionális kihívások, például a bérléskezelési modulok pontos leképezése a szoftverben, szintén komoly figyelmet igényeltek. A szakirodalom kiemeli, hogy a komplex üzleti folyamatok modellezése iteratív tesztelés és visszacsatolás nélkül gyakran vezet félreértésekhez és hibákhoz [2, 4]. Saját értékelésem szerint a moduláris fejlesztés és az ismételt tesztelés lehetővé tette, hogy a rendszer logikája a vállalati igényekhez igazodjon, miközben a hibák gyorsan lokalizálhatók és javíthatók voltak.

Az időmenedzsment kritikus tényező volt, mivel a projektet egyedül végeztem. A napi és heti ütemtervek készítése, valamint a feladatok priorizálása lehetővé tette a munka és a projekt előrehaladásának kiegyensúlyozását. A magyar szakirodalom szerint az időbeosztás és a feladatpriorizálás kulcsfontosságú a projekt határidőn belüli teljesítéséhez és a kiégés elkerüléséhez [1, 5]. Saját tapasztalatom szerint a tervezett és dokumentált ütemezés hozzájárult a hatékony munkavégzéshez és a stressz csökkentéséhez.

A tesztelés és hibajavítás terén a moduláris stratégia alkalmazása biztosította, hogy minden komponens külön-külön ellenőrzésen essen át, majd a teljes rendszer integrációját is alaposan teszteltem. A szakirodalom hangsúlyozza a folyamatos integráció és a moduláris tesztelés szerepét a hibák korai felismerésében és a rendszer megbízhatóságának növelésében [2, 4]. Saját értékelésem szerint ez a megközelítés lehetővé tette a hibák gyors javítását és a stabil rendszer kialakítását, minimalizálva a későbbi problémákat.

Összességében a problémák kezelésére alkalmazott stratégiai megközelítés biztosította a projekt sikeres előrehaladását, miközben a rendszer stabil, funkcionális és a vállalat igényeinek megfelelő maradt.

4.4. Tanulságok a kockázatkezelésből

A projekt során szerzett tapasztalatok világosan rámutattak arra, hogy a kockázatkezelés elengedhetetlen eleme az önálló fejlesztéseknek. A problémák előrejelzése és a priorizálás nem csupán a projekt sikerességét növeli, hanem lehetővé teszi a fejlesztési folyamat hatékony szervezését és a kritikus hibák minimalizálását [5, 1, 2]. Az előzetesen kialakított kockázatmátrix segített azonosítani a legkritikusabb területeket, így a figyel-

met a legfontosabb problémákra tudtam összpontosítani, miközben a kisebb kockázatok későbbi kezelésre vártak.

A rendszeres tesztelés és a dokumentáció készítése szintén kulcsfontosságú volt. A szakirodalom hangsúlyozza, hogy a moduláris tesztelés, az iteratív javítások és a részletes dokumentáció lehetővé teszik a hibák gyors felismerését és hatékony javítását, ami jelentősen csökkenti a rendszerhibák előfordulását [4, 2]. Saját tapasztalatom szerint a folyamatos tesztelés és dokumentáció nemcsak a rendszer stabilitását biztosította, hanem a projekt átláthatóságát is növelte, így a későbbi bővítések is könnyebben megvalósíthatók voltak.

Az időmenedzsment és a feladatpriorizálás szintén kiemelt jelentőségűnek bizonyult. Az önálló fejlesztés során a napi és heti ütemtervek, valamint a feladatok rangsorolása lehetővé tette a projekt folyamatos előrehaladását anélkül, hogy a munkavégzés minősége vagy a határidők veszélybe kerültek volna [1, 5]. Saját értékelésem szerint a strukturált időbeosztás és a fókuszált feladatvégzés elengedhetetlen a sikeres, önálló projektvégrehajtáshoz.

Összességében a kockázatkezelés alkalmazása nemcsak a rendszer megbízhatóságát és funkcionalitását biztosította, hanem hozzájárult a projekt menedzsmenti tapasztalatok megszerzéséhez is. A megszerzett tanulságok alapján a jövőbeli projektek során a kockázatkezelés tudatos, iteratív és folyamatos alkalmazása elengedhetetlen a hatékony és eredményes fejlesztéshez.

5. fejezet

Tanulságok és szakmai összegzés

A **TeDeRMS** projekt nem csupán egy vállalati bérléskezelő rendszer létrehozását jelentette, hanem egy teljesen önálló projektmenedzsment gyakorlatot is. A fejlesztés során szerzett tapasztalatok világosan megmutatták, hogy a siker kulcsa a tudatos tervezés, a kockázatok proaktív kezelése és a moduláris, skálázható fejlesztési megközelítés.

5.1. Projektmenedzsment tapasztalatok

A projekt során szerzett tapasztalataim alapján a részletes tervezés és az ütemezés kulcsfontosságú volt az önálló fejlesztés sikeréhez. A mérföldkövek alkalmazása segített abban, hogy a kritikus feladatokra mindig a megfelelő időben tudjak koncentrálni, és ne veszítsem el a fókuszt a rendszer stabil és hibamentes működésén. Saját tapasztalatom szerint, ha a feladatokat nem a mérföldkövekhez igazítottam volna, gyakran előfordult volna, hogy a kisebb, kevésbé fontos részletek lekötik az időt és energiát, így a projekt határideje veszélybe került volna.

A kockázatok előrejelzése és figyelése rendkívül hasznosnak bizonyult. A projekt során a kritikus problémákat igyekeztem még a bekövetkezésük előtt felismerni, így minimalizálni tudtam az előre nem látható akadályok okozta fennakadásokat. Ez a proaktív megközelítés nemcsak a projekt menetét tette gördülékenyebbé, hanem a saját döntéshozatali képességemet is fejlesztette, hiszen mindig tisztában voltam a legfontosabb teendőkkel és a várható kihívásokkal.

Az önálló erőforrás-menedzsment terén az idő és az eszközök tudatos optimalizálása vált a siker egyik kulcstényezőjévé. A napi és heti ütemtervek betartása, valamint a feladatok rangsorolása biztosította, hogy a projekt folyamatosan haladjon, anélkül hogy túlterhelt vagy demotivált lettem volna. Ebből a szempontból úgy érzem, hogy a projekt nem csupán a technikai készségeimet, hanem a saját felelősségvállalásomat és önálló munkavégzési képességemet is jelentősen fejlesztette.

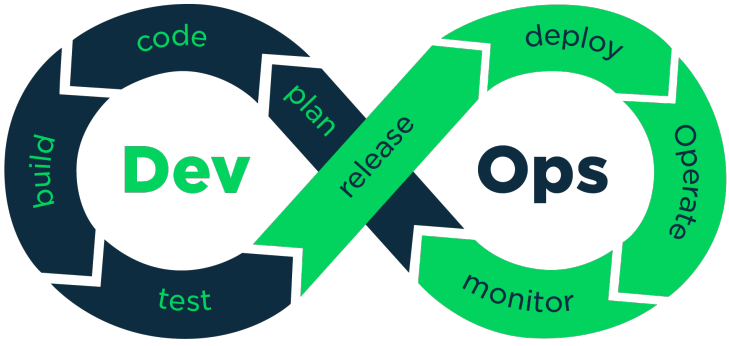
Összességében úgy látom, hogy a tudatos tervezés, a proaktív kockázatkezelés és az önálló erőforrás-menedzsment integrált alkalmazása nemcsak a projekt céljainak elérését segítette, hanem hozzájárult a saját fejlődésemhez is. A projekt során szerzett tapasztalatok alapján a jövőbeni önálló fejlesztések során ezek az elvek elengedhetetlenek a hatékony és sikeres munkavégzéshez.

5.2. Fejlesztési és technológiai tanulságok

A projekt során szerzett tapasztalataim alapján a moduláris és skálázható architektúra kialakítása kulcsfontosságú volt. A rendszer moduláris felépítése lehetővé tette, hogy a különböző funkcionális egységeket egymástól függetlenül fejlesszem, teszteljem és javítsam, ami jelentősen gyorsította a hibajavítási folyamatokat, és könnyebbé tette a későbbi bővítéseket. Tapasztalatom szerint ez a rugalmasság nem csupán a fejlesztői munka hatékonyságát növelte, hanem hosszú távon a vállalat igényeinek kiszolgálását is biztosítja.

A folyamatos tesztelés és a részletes dokumentáció szintén meghatározó tényezőnek bizonyult. A moduláris tesztelési stratégia lehetővé tette, hogy a hibákat gyorsan lokalizáljam és javítsam, miközben a rendszer teljes integritása megőrződött. A dokumentáció elkészítése nem csupán a rendszer áttekinthetőségét segítette, hanem a felhasználók számára is könnyebbé tette az eligazodást, ami a rendszer elfogadottságát és a hatékony működést növelte.

A technológiai integráció terén a PHP/Twig backend és a Docker alapú konténerizálás kombinációja jelentős előnyöket biztosított. A konténerizált környezet stabilitást és rugalmasságot biztosított, miközben a rendszer telepítése és frissítése gyorsan elvégezhető volt. Saját tapasztalatom szerint ez a megoldás csökkentette a konfigurációs problémák és futtatási hibák előfordulását, valamint lehetővé tette, hogy a vállalat az új rendszert azonnal, biztonságosan és hatékonyan használja.



5.1. ábra. DevOps szemlélet a projektben

Ábrák jegyzéke

Irodalomjegyzék

[1] Kovács Gábor: *Projektmenedzsment a gyakorlatban*. Budapest, 2016, Typotex Kiadó. ISBN 9789632798201.

[2] Szalay Imre: *Projektmenedzsment alapok: módszertan és gyakorlat*. Budapest, 2018, Budapesti Műszaki és Gazdaságtudományi Egyetem. ISBN 9789633133377.

[3] Simon István: *Modern projektmenedzsment módszertanok*. Budapest, 2015, Complex Kiadó. ISBN 9789632972435.

[4] Kaposi József: *Agilis és hagyományos projektmenedzsment: elmélet és gyakorlat*. Budapest, 2019, Budapesti Corvinus Egyetem Kiadó. ISBN 9789635301293.

[5] Hajdu Miklós: *Projektmenedzsment*. Budapest, 2014, Akadémiai Kiadó. ISBN 9789634540181.