



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR



SZAKDOLGOZAT

OE-NIK
2023

Hallgató neve:
Hallgató törzskönyvi száma:

Réthy Bálint
T-002237/FI12904

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertervezés és -fejlesztés Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Réthy Bálint**
Törzskönyvi száma: **T/002237/FI12904/N**
Neptun kódja: 

A dolgozat címe:

**Vállalatirányítási információs rendszer kis- és középvállalkozások
számára**
Enterprise resource planning for small and medium-sized companies

Intézményi konzulens: **Balázs Elemér**
Külső konzulens:

Beadási határidő: **2022. május 15.**

A záróvizsga tárgyai: **Számítógép architektúrák**
Szoftvertechnológia

A feladat

Tervezzon meg és készítsen el egy olyan vállalat-menedzsment rendszert, amely hatékonyan támogatja kis- és középvállalatok általános ügymenetét. Az összefüggő adatok beviteléhez határozzon meg egy üzleti folyamatot, melyet részletezzon a dokumentumban. A rendszer egy webes alkalmazás legyen, melyre megfelelő autentikációt követően lehessen belépni. Alakítson ki különböző jogosultsági szinteket, amelyek hozzárendelhetők legyenek a megfelelő alkalmazottakhoz, külső partnerekhez, ügyfelekhez.

Alakítson ki egy olyan szolgáltatást, mely a rendelkezésre álló belső adatok alapján képes projektütemezést biztosítani a vállalat számára.

A feladat megvalósításakor alkalmazza az ASP.NET MVC technológiát. Az adatok rendszerezett tárolására adatbázist használjon. A rendszer tervezése és megvalósítása során törekedjen a szoftvertechnológia modern elveinek kialakítására.

A dolgozatnak tartalmaznia kell:

- a feladat leírását,
- a kapcsolódó irodalom részletes feldolgozását,
- hasonló rendszerek bemutatását és jellemzését,
- az alkalmazás teljeskörű tervezését és a kiválasztott technológiák indoklását,
- a támogatott üzleti folyamat ismertetését,
- projektütemező szolgáltatás kialakításának dokumentálását,
- az alkalmazás tesztelési jegyzőkönyvét és eredmények bemutatását,
- továbbfejlesztési lehetőségeket,
- a dokumentációt, a programot, a szükséges inputadatokat, valamint a rendszert bemutató honlapot és prezentációt CD mellékleten.



Vámossy Zoltán

Dr. Vámossy Zoltán
intézetigazgató

A szakdolgozat elévülésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022.12.11.

.....
hallgató aláírása



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve:

RETHY BALINT

Neptun Kód:

[REDACTED]

Tagozat:

NAPPALI

Telefon:

Levelezési cím (pl.: lakcím):

[REDACTED]

Szakdolgozat / Diplomamunka⁶ címe magyarul:

VÁLLALATI RENDSZER INFORMATIKAI RENDSZER KIS- ÉS KÖZÉPVÁLLALKOZÁSOK SZÁMÁRA

Szakdolgozat / Diplomamunka⁷ címe angolul:

ENTERPRICE RESOURCE PLANNING FOR SMALL AND MEDIUM-SIZED COMPANIES

Intézményi konzulens:

BALÁZS ELEMER

Külső konzulens:

[REDACTED]

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.02.07.	PROJEKT INICIALIZÁLO MEGBESZÉLÉS	[Signature]
2.	2022.02.28.	KÖVETELMÉNYEK VÉGLEGESÍTÉSE	[Signature]
3.	2022.04.18.	AZ ÁTTEKINTÉST IRÓDALOM VÉLEMÉNYEZÉSE	[Signature]
4.	2022.05.02.	RENDSZERTERV ÁTBESZÉLÉSE	[Signature]

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4⁸ tantárgy követelményét teljesítette, beszámolóra / védésre⁹bocsátható.

[Signature]

Intézményi konzulens

Budapest, 2022.05.06.

⁶ Megfelelő aláhúzendő!

⁷ Megfelelő aláhúzendő!

⁸ Megfelelő aláhúzendő!

⁹ Megfelelő aláhúzendő!



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve:

RETHY BALINT

Neptun Kód:

[REDACTED]

Tagozat:

NAPPALI

Telefon:

Levelezési cím (pl.: lakcím):

[REDACTED]

Szakdolgozat / Diplomamunka⁵ címe magyarul:

VÁLLALATI RÁNYÍTÁSI INFORMÁCIÓS RENDSZER KIS-ÉSKÖZEP VÁLLALKOZÁSOK SZÁMÁRA

Szakdolgozat / Diplomamunka⁶ címe angolul:

ENTERPRICE RESOURCE PLANNING FOR SMALL AND MEDIUM-SIZED COMPANIES

Intézményi konzulens:

BALÁZS ELEMEŰ

Külső konzulens:

[REDACTED]

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022. 09. 05.	ALKALMAZÁS VÁZÁNAK ISMERTETÉSE	[Signature]
2.	2022. 09. 23.	TERHELÉS ELOSZTÁSI LEHETŐSÉGEK ÁTBESZÉLÉSE	[Signature]
3.	2022. 11. 04.	AZ ELÉRT EREDMÉNYEK ISMERTETÉSE	[Signature]
4.	2022. 11. 25.	PROJEKT ZÁRÁSSAL KAPCSOLATOS TEENDŐK MEGBESZÉLÉSE	[Signature]

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat / Szakdolgozat I. / Szakdolgozat II. / Diplomamunka II. / Diplomamunka III. / Diplomamunka IV. / Projektlabor 2. / Projektlabor 3. / Záródolgozati projekt⁷ tantárgy követelményét teljesítette, beszámolóra / védésre⁸bocsátható.

Javasolt érdemjegy:⁴.....

[Signature]

Intézményi konzulens

Budapest, 20. 22. 11. 28.

⁵ Megfelelő aláhúzendő!

⁶ Megfelelő aláhúzendő!

⁷ Megfelelő aláhúzendő!

⁸ Megfelelő aláhúzendő!

TARTALOMJEGYZÉK

1.	Bevezetés	3
1.1.	Az ERP-ről általánosságban.....	3
1.2.	Elterjedését hátráltató tényezők	4
1.3.	Előnyök összegzése.....	4
2.	Irodalomkutatás	5
2.1.	Egy ERP bevezetésének lépései	5
2.1.1.	Kutatás és tervezés	5
2.1.2.	Tervezés	5
2.1.3.	Konfigurálás és fejlesztés	6
2.1.4.	Tesztelés.....	6
2.1.5.	Indítás és telepítés	6
2.1.6.	Támogatás és karbantartás	7
2.2.	Hasonló rendszerek bemutatása	7
2.2.1.	Legelterjedtebb ERP-k.....	7
2.2.2.	Ingyenesen elérhető ERP-k.....	10
2.3.	Terheléselosztás	13
2.3.1.	Terheléselosztó algoritmusok	15
2.3.2.	Hardveres és szoftveres terheléselosztás összehasonlítása	17
2.3.3.	Parancs-lekérdezés felelősség elkülönítése	18
3.	Tervezés	20
4.	A megoldási módszer kiválasztása, a választás indoklása.....	25
5.	Megvalósítás	26
5.1.	Rétegelt alkalmazás felépítése	26
5.2.	Interfészek, entitások definiálása	27
5.3.	Létrejött adatbázis táblák	28
5.4.	Felületek és funkciók megvalósítása.....	29
5.4.1.	Törzsadatok.....	30
5.4.2.	Pénzügyek.....	36
5.4.3.	Napi tevékenységek	37

5.5.	Jogosultságok áttekintése	40
5.6.	Kimutatások készítése	40
6.	Tesztelés.....	42
6.1.	Egységtesztek.....	42
6.1.1.	Minden entitás esetén megvalósított vizsgálatok.....	42
6.1.2.	Egyedi vizsgálatok entitás típusonként.....	42
6.2.	Felületi tesztek	43
6.3.	Stressztesztek	44
7.	Továbbfejlesztési lehetőségek	46
8.	Összefoglalás	47
9.	Abstract.....	48
10.	Irodalomjegyzék	49

1. BEVEZETÉS

A vállalkozások gyakran minimális szoftveres támogatással kezdik meg munkájukat, amely a későbbiekben is jellemző lehet. Egy szervezet növekedése magával vonja az egyre növekvő szükséges adminisztrációkat is, amelyek segítenek áttekinteni a bevételek, kiadások, ráfordítások, humán erőforrások és egyéb üzleti folyamatok együttesét. Ezen feladatok gyakran ismétlődőek. Áttekintésükre Excel táblázatokat, jegyzeteket, esetleg papír alapú megoldást választanak. Az igényekre szabott szoftveres támogatás hiánya nem csak az új cégek, vállalkozások esetén figyelhető meg, hanem azon vállalkozások esetében is, amelyek több tíz éve működnek és már kialakult módszereik vannak. Ezen kívül gyakori, hogy elavult, akár DOS (*Disk Operating System*) alapú vállalkozásra szabott alkalmazást használnak. Egy modern technológiával elkészített vállalatirányítási információs rendszer – továbbiakban ERP (*Enterprise Resource Planning*) - jelentősen növelheti a vállalkozások hatékonyságát, áttekinthetőségét és csökkentheti az adminisztrációhoz szükséges idő ráfordítást.

1.1. Az ERP-ről általánosságban

A vállalatirányítási információs rendszer a vállalat környezetére, belső működésére és a vállalat és a környezet tranzakcióira vonatkozó információk beszerzését, feldolgozását, tárolását végző személyek, tevékenységek, funkciók ellátását lehetővé tevő hardverek és szoftverek együttese. Szigorúan véve az ERP a vállalati erőforrás-tervező. [1]

Több mint 40 éve léteznek ERP-k. Ez idő alatt több tízezer cég vezette használatukat. Emberek milliói használják a mindennapi munkájukhoz megkönnyítve és felgyorsítva az adminisztrációs folyamatokat. Az ERP két fő jellemzője az adatintegráció és a legjobb gyakorlati folyamatok (*best practices*) alkalmazása. Az adatintegráció azt jelenti, hogy az adatokat elegendő egyszer bevinni a rendszerbe, ezután elérhetőek mindenütt. Jellemzően ezeket az adatokat párhuzamosan több helyen tárolták, míg az ERP bevezetése után a szervezeten belül ezek az adatok egy központi adattárból elérhetőek. Az adatokhoz hozzáférést lehet biztosítani a kívánt felhasználóknak, az adatok bevitele, frissítése pedig történhet automatikusan vagy jogosultságok, szerepkörök alapján manuálisan. Egy modern ERP rendszer számos gyakorlati folyamattal rendelkezik. Ezeket bevezethetik az ERP használatának megkezdésével vagy fokozatosan kiegészíthetik a jelenlegi rendszerüket, ezzel növelve a hatékonyságukat. [2]

Az adatbiztonság és az illetéktelen hozzáférés megakadályozása elengedhetetlen szempont, valamint a kezelő személyzet megfelelő oktatása a program használatához, ellenkező esetben a kívánt átláthatóság, hatékonyság növelés és biztonság nem garantált.

1.2. Elterjedését hátráltató tényezők

Mielőtt egy szervezet egy ERP használata mellett döntene meg kell vizsgálni, hogy valóban szükség van-e ERP-re. Egy ERP használata akkor hozza meg a várt pozitív hatást, ha egyértelműen definiálva vannak a szervezet folyamatai, ha tudják, hogy milyen problémák megoldására szeretnék használni, valamint kellő finansziális és humán erőforrás áll rendelkezésre. Kezdetben sok idő szükséges a rendszert kezelő személyzet számára, hogy megismerje és megfelelően kezelje az alkalmazást. Gyakori probléma, hogy ragaszkodnak a régi, jól bevált módszerekhez és nem szeretnék a folyamataikat egy ERP rendszeren keresztül menedzselni. [3]

Léteznek ERP-k, melyek általános felhasználói igényt elégítenek ki, valamint léteznek olyanok, melyek egyedi folyamatok és módszerek miatt speciális fejlesztést igényelnek. Mindkét esetben lehet egyszeri költség és/vagy folyamatos költség (*pl. havidíjas termék*). Egyedi igényekre szabott fejlesztés esetén az ERP lefejlesztése jóval nagyobb költséget jelent, mint egy már kész alkalmazáshoz való hozzáférés vásárlása. Az egyszeri bekerülési költség is kiegészülhet később egyéb költségekkel, pl. alkalmazás támogatás, továbbfejlesztés. Általánosságban 5-10 évig lehet egy ERP-vel számolni mielőtt elavulttá válik, mivel ezalatt a folyamatok, a szervezet, az ügyfelek is teljesen átalakulhatnak. [4]

1.3. Előnyök összegzése

Egy vállalkozás számára az ERP bevezetése óriási előnyökkel járhat, főleg, ha a szervezet növekedése a cél. Egy jó vállalatirányítási rendszer moduláris, rendelkezik számos általános célú funkcióval, amelyek igény szerint bekapcsolhatóak, valamint kiegészíthetők egyedi modulokkal, hogy vállalkozásra szabott alkalmazás legyen a végeredmény. Egy közös alkalmazást tud használni például a könyvelés, a logisztika, a menedzsment, a termelés és a HR (*Human Resources*). Jobb rálátást biztosít az üzleti folyamatokra, költségekre és bevételekre. Különböző kimutatásokat tud végezni és segíti a projekt, költség becslést is. Az eleinte befektetett plusz munka az adminisztráció és az alkalmazás megismerése hamar növeli a teljesítményt az ismétlődő feladatok automatizálásával és az adatok újrafelhasználásával. Csökkennek a humán erőforrás költségek is. Az adatok egy központi adatbázisban tárolódnak, innen érik el a különböző ágakat. A validációknak köszönhetően garantálható az adathelyesség és növeli az előírásoknak való megfelelést is, így a szabványoknak, auditkritériumoknak is könnyebb megfelelni. Megfelelő autentikációval, autorizációval és architektúrával az adatbiztonság is elérhető. A műveletek naplózásával visszakövethető, hogy melyik felhasználó milyen interakciókat végzett az alkalmazásban. [2] [5]

2. IRODALOMKUTATÁS

Egy ERP bevezetésének jól elkülöníthető lépései vannak, azonban a lépésekben szereplő tevékenységek lehetnek átfedésben. Az alábbi fejezet ezt fejt ki részletesen, valamint bemutatja a legelterjedtebb ERP rendszereket és az ingyenes alternatívákat. Ezt követően a fejezet a terheléelosztás lehetséges megvalósításait taglalja.

2.1. Egy ERP bevezetésének lépései

Az ERP bevezetésének lépései a [6]-os forrás feldolgozásával készült.

Egy vállalkozás számára egy ERP bevezetése hónapokig is tarthat, érdemes ezzel számolni. Mielőtt azonban a bevezetés mellett döntenének érdemes átgondolni, hogy valóban az ERP jelenti-e a megoldást. Meg kell határozni a célokat, átgondolni az üzleti folyamatokat és hogy milyen elvárások vannak az ERP felé. Elkötelezettséget igényel mind a vezetés, mind a munkatársak részéről az alkalmazkodás, tanulás és tesztelés terén, valamint fel kell mérni, hogy a bevezetés vagy fejlesztés felmerülő költségei rendelkezésre állnak-e. Általános esetben a bevezetést 6 részre lehet felosztani, melyek a következő alfejezetekben lesznek kifejtve.

2.1.1. Kutatás és tervezés

A bevezetés folyamatához szükséges egy vezető kiválasztása. Ő felel a bevezetés koordinálásáért, az oktatásért, valamint a kommunikációért a fejlesztő cég felé. Amennyiben ilyen munkatárs nem áll rendelkezésre, akkor lehetőség van ERP tanácsadót is fogadni.

Ebben a lépésben kell megfogalmazni az igényeket, elvárásokat, valamint a nem hatékony és ismétlődő folyamatokat. Fontos a megfelelő támogatás a vezetés részéről és az erőforrások biztosítása, továbbá ki kell jelölni minél több -ideális esetben minden- munkatársat az alkalmazás tesztelésére.

Hogyha megszületett a döntés az ERP bevezetéséről, akkor piackutatást kell végezni. Számos elérhető ERP közül lehet választani, különböző célokra más és más választás lehet ideális. A legtöbb ERP elérhető webes felületen és mobil eszközökről is. Célszerű minél modernebb technológiát választani, akár felhő alapú megoldás is szóba jöhet a saját infrastruktúra kiváltása helyett. Ezeket az alkalmazásokat ki lehet egészíteni egyedi funkciókkal, vagy lehet saját alkalmazás fejlesztést is igényelni, azonban ez sokkal költségesebb és jóval több időt vesz igénybe.

2.1.2. Tervezés

A tervezés már elkezdődött az előző lépésben, azonban itt történik a részletes, minden területre kiterjedő tervezés. Az ERP használata megkövetelheti az üzleti folyamatok

átalakítását, hogy kompatibilis legyen az alkalmazással. Fontos, hogy a vállalkozás minden területéről, minél több munkatárs be legyen vonva a tervezésbe, mert minél szélesebb ismeretek vannak felhasználva, annál jobban tudja kiszolgálni a rendszer az összes ágazatot. A munkatársak bevonása a tervezési folyamatba növeli az elfogadást és a használatának hajlandóságát.

Meg kell határozni a folyamatokat, majd a hiányosságokat is, és ezekre is megoldást kell találni, amelyek a legtöbb esetben egyedi modulok fejlesztésével orvosolhatóak.

2.1.3. Konfigurálás és fejlesztés

A meghatározott igények és célok teljesítése után megkezdődhet az alkalmazás konfigurációja. Egyedi igények alapján további fejlesztések lehetnek indokoltak. Más szoftverek egyidejű használata esetén integrálni kell a céges környezetbe, például számlázó, könyvelő programmal összekötni.

Már a fejlesztés alatt, az első teszt verziótól kezdve tesztelhetik a munkatársak és ez idő alatt már érdemes oktatást is tartani. Ebben a lépésben kezdődhet az adatmigráció (*régi rendszerből adatok átvitele az új rendszerbe*) is. Ez gyakran sok munkát igényel, le kell tisztázni mely adatokat szükséges átvinni a rendszerbe. A duplikációkat és felesleges adatokat ki kell szűrni, valamint az új rendszernek megfelelő formára kell hozni az adatokat.

2.1.4. Tesztelés

Ideális esetben a tesztelés is elkezdődik már az előző lépésben. A rendszer minden modulját szükséges alaposan tesztelni, valamint végponttól végpontig tesztelni (*end-to-end testing*). A modulokat a tesztelés visszajelzései alapján lehet módosítani vagy az új igényeknek megfelelően testre szabni.

Ebben a lépésben szükséges a migrált adatok ellenőrzése is. Amennyiben még nem kezdődött meg, akkor mielőbb szükséges a munkatársak oktatása. Ez általában szöveges oktatóanyag formájában valósul meg, amely kiegészülhet képi, videós tananyaggal is. Emellett a személyes oktatás is jellemző.

2.1.5. Indítás és telepítés

Ebben a lépésben élesítik a rendszert. Számítani kell fennakadásokra, hibákra, mivel rengeteg komponensből áll az élesítés folyamata. A projekt bevezetéséért felelős kollégának és a fejlesztőknek is rendelkezésre kell állni, hogy a felmerülő problémákat mielőbb megoldják. Hogyha valamilyen okból -pl.: céges szabályzat- még nem történt meg az adatmigráció, vagy a legutóbbi migráció óta újabb adatok keletkeztek, akkor ezeket is pótolni kell.

2.1.6. Támogatás és karbantartás

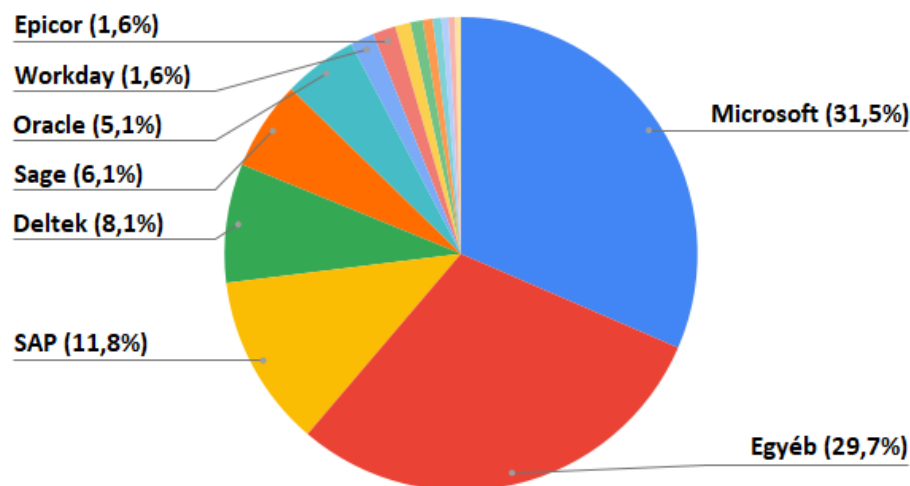
A támogatás és a továbbfejlesztés elengedhetetlen egy hatékony ERP esetén, valamint a rendszer felhasználói is elégedettebbek lesznek és szívesebben használják. A továbbfejlesztésnek köszönhetően folyamatosan fejlődhet az ERP, valamint követheti a vállalkozás folyamatait és a szervezet növekedését. Általában a bevezetésért felelős személy felel a támogatás és a továbbfejlesztés koordinálásáért, valamint ő kommunikálja a fejlesztő csapat felé a munkatársak visszajelzéseit.

2.2. Hasonló rendszerek bemutatása

Az alábbi fejezet a piaci részesedés alapján legelterjedtebb ERP-ket mutatja be, a lista elején szereplőket részletesen kifejtve. Ezután az ingyenesen elérhető rendszereket mutatom be, amelyek többnyire tartalmaznak fizetős szolgáltatásokat is, de ezek egyből a bevezetés utáni időszakban nem feltétlen szükségesek.

2.2.1. Legelterjedtebb ERP-k

ERP választásakor számos opció ideális lehet, nem feltétlenül szükséges vagy jobb egyedi fejlesztésű rendszer mellett dönteni. A következőkben a legnagyobb ERP-ket tekinthetjük át. Az 1. ábrán a legnagyobb ERP-k piaci eloszlása tekinthető meg a 2022-es állapotuk alapján.



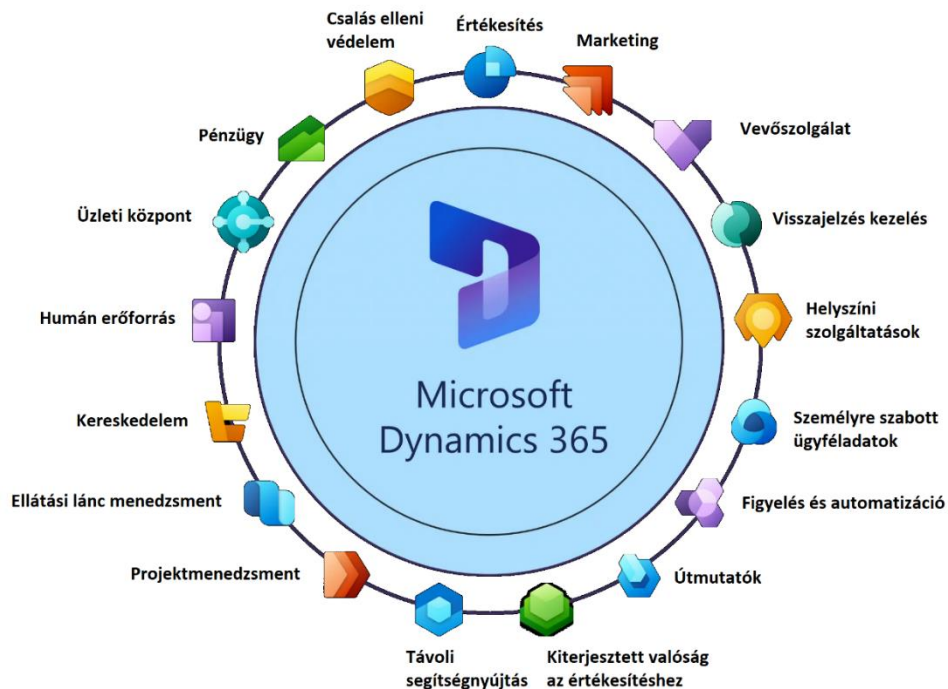
1. ábra - ERP-k piaci eloszlása 2022-ben [7]

Microsoft

A Microsoft piaci részesedése 31,5%. Több éve vezető az ERP-k terén a Dynamics 365 termékük, amely lefedi többek között a HR, termelés, beszerzés, működési folyamatok és pénzügy ágazatokat. Felhő alapú technológia, amely összekapcsolja a modulokat egy alkalmazásba és együttműködik más Microsoft termékekkel, mint pl.: Outlook, Office

365. Nagy előnye, hogy egyben ERP és CRM (*Customer Relationship Management – ügyfélkapcsolat kezelő*) is és nagyon jól testesztelhető. Elsősorban kis- és középvállalkozások számára ajánlott, de használja sok nagyobb szervezet is. Az alkalmazottak számától függően eltérő árazás alapján lehet hozzáférni a rendszerhez. [7] [8]

A piac legnagyobb szeletét birtokló Microsoft Dynamics 365 számos tartalmat biztosít az ERP rendszerében, mely a 2. ábrán tekinthető meg.



2. ábra - Microsoft Dynamics 365 szolgáltatások [8]

SAP

Az SAP a Systems, Application and Products (*Rendszerek, Alkalmazások, Termékek*) rövidítése. Sokáig vezető szerepet töltött be a vállalatirányítási rendszerek terén az 1972-es indulása óta, most 11,8% a piaci részesedése. Az SAP Business One termékük kifejezetten olyan kisvállalatok számára készült, amelyek szeretnének növekedni. Egy közös rendszerbe van integrálva a könyvelés, a gyártás, a leltár kezelés, a raktárkezelés, a beszerzés, valamint a humán erőforrás menedzsment. A Microsoft Dynamics 365-höz hasonlóan ez is CRM rendszer is. Felhő alapú szolgáltatás, de igény esetén a vállalkozások dönthetnek a külső vagy lokális futtatás mellett is. Nagyon jó a megítélése a megbízható fejlesztő cég, a megbízható alkalmazás, valamint a kedvező ára miatt. Emellett a teljesítménye is kiemelkedő. A kimutatásoknak köszönhetően növekedésre ösztönzik a vállalkozásokat és támogatják ezt a folyamatot. [7] [9]

A 3. ábrán az SAP ERP modulok tekinthetők meg.



3. ábra - SAP ERP modulok [9]

Deltek

A piaci részesedésből 8,1%-ot tesz ki. Vezető ERP a projekt alapon működő vállalkozások számára. 1983-ban indult, főleg állami közbeszerzéseket megvalósító vállalkozásokra fókuszált. Hatékonyan tudta követni a projekt alapú költségeket, amely akkoriban nem sok konkurens ERP-nek sikerült. Így alakult ki a Deltek Government Contracting System (*Deltek kormányzati szerződéskötési rendszer*). [7]

Modern ERP alkalmazásuk a Deltek Vantagepoint, amely projekt alapú könyvelést biztosít mérnökök, építészek és tanácsadó cégek számára. Hatékonyan támogatja a projekt menedzsmentet, személyre szabott kimutatásokkal teszi átláthatóvá az idő és költség ráfordítás követését, valamint az erőforrás menedzsmentet. Az ERP mellett CRM is és szolgáltatás automatizálást (*PSA – Professional Service Automation*) is kínál. Integrálható számos szoftverrel, mint pl.: Microsoft Office 365, SharePoint, Dynamics CRM, valamint elérhető angol, német, spanyol, portugál és holland nyelven. [7] [10]

A 4. ábra a Deltek Vantagepoint ERP szolgáltatásait szemlélteti.



4. ábra - Deltek Vantagepoint szolgáltatások [10]

Sage

A piaci részesedés 6,1%-ért felelős. A számviteli megoldások és az ERP-k terén globális megoldás. Elsősorban a kisvállalkozásokra van specializálódva. Kiemelkedő az automatizálás és a döntéstámogatás terén. [7]

Modern alkalmazásai többek között a Sage 300, Sage 500 és a Sage Business Cloud Accounting. A Sage 300 erősebb a személyre szabhatóság, a biztonság, a kimutatások, a főkönyv, a bérszámfejtés és a könyvelés terén, míg a Business Cloud Accounting a készletkezelésben és a rendeléskezelésben ért el magasabb pontszámokat, valamint elérhető próbaverzió is. [11]

A Sage 300 és a Sage 500 nagyon hasonlóak, de a Sage 500 kevésbé elterjedt, mivel általában a több mint 1000 alkalmazottal rendelkező cégek használják. A Sage 300 elérhető számos nyelven, míg a Sage 500 csak angol nyelven. [12]

Oracle

A piaci részesedés 5,1%-át teszi ki. Az Oracle a világ egyik legnagyobb szoftver vállalata, amely adatbázis szolgáltatásokat, felhő alapú szoftvereket és ERP-ket biztosít. Vállalatirányítási szoftvereik hatékonyan támogatják a humántőke-menedzsmentet (*HCM – Human Capital Management*), az ügyfélkapcsolat-kezelést, valamint az ellátási lánc menedzsmentet. Megjelenik a gépi tanulás (*Machine Learning*) is, amely a mesterséges intelligencia része. Matematikai adatmodelleket és nagy adatmennyiséget használ, folyamatosan fejlődik és alkalmazkodik a változó igényekhez. Ezek következményeképp pontosabb becslést tud adni és feltárja a jelentős hibákat. Szorgalmazza az üzleti folyamatok automatizálását, a gyakori feladatokat leegyszerűsíti és felgyorsítja. A mesterséges intelligencia akár a tranzakciók 96%-át automatizálhatja. [7] [13]

2.2.2. Ingyenesen elérhető ERP-k

Sok ERP szoftver ingyenesen is elérhető. Ezek nagy előnye, hogy a vállalkozásoknak lehetőségük van kipróbálni ERP rendszereket. Eldönthetik, hogy valóban szükségük van-e rá, segíti-e az üzleti folyamataik menedzselését és hogy növelte-e a hatékonyságot. Az ingyenesen elérhető szoftvereknél is lehetnek rejtett költségek. A bevezetés és karbantartás ugyanúgy szükséges, mint egy fizetős változatnál. [14]

Az ingyenesen elérhető vállalatirányítási rendszerek egy része nyílt forráskódú (*Open Source*), amelynek előnye, hogy szabadon használhatóak és az egyedi igényeknek megfelelően módosíthatóak, továbbfejleszthetőek. Annak ellenére, hogy az alap szoftver nyílt forráskódú, lehetnek olyan modulok, amelyek csak fizetés ellenében érhetőek el. [15]

A másik részüket cégek fejlesztik, itt legtöbbször az ingyenes verzió kiegészíthető prémium funkciókkal, szolgáltatásokkal vagy valamilyen korlátozással vehető igénybe díjmentesen, mint pl.: csak saját szerveren futtatva (*self hosting*) vagy csak egy felhasználóval használva. Ahogy a cégek nőnek és látják a vállalatirányítási szoftver hasznosságát egyre inkább elgondolkoznak, hogy igénybe vegyék a fizetős szolgáltatásokat, amelyekkel szélesebb körben tudják támogatni üzletmenetüket, valamint jelentős mennyiségű költséget és időt is spórolhatnak vele. [14] [15]

Az ingyenes és fizetős szolgáltatásoknál érdemes megemlíteni a próbaváltozatokat (*trial version*), amelyek lehetőséget biztosítanak a cégeknek egy szoftver kipróbálására egy határozott ideig (*általában 2 hét vagy 1 hónap*), majd, ha sikerült elnyerni a tetszésüket, akkor a további használat díjköteles. [15]

A következőkben néhány ingyenesen elérhető ERP lesz bemutatva.

Bitrix24

A Bitrix24 egy felsőkategóriás ERP, amely kiemelkedően jó felhasználói értékelésekkel rendelkezik. Több, mint 35 együttműködési eszközzel támogatja a csoportokat a kívánt cél elérése érdekében. A felhasználóknak lehetőségük van feladat- és projekt menedzsment eszközöket használni. A Bitrix24 ERP egyben CRM is, valamint a cégen belüli kommunikációban is számos funkcióval bír, pl.: csevegés, nagyfelbontású videóhívások, hívások alatt élő beszéd feliratozás, valamint közös adattár és tudástár az információk megosztására. [15]

Nagy hangsúlyt fektetnek a biztonságra. Kliensenként külön tárolási rétegeken adat izolációt használnak. Az adatátvitelhez SSL (*Secure Socket Layer*) tanúsítványt használnak, amely egy titkosított csatorna a szerver és a kliens között. A felhasználói adatok megfelelnek az általános adatvédelmi rendeletnek (*GDPR – General Data Protection Regulation*). Adatközpontjaik megfelelnek a SAS 70 Type II szabványnak, amely a legszélesebb körben elfogadott audit standard, valamint a Safe Harbor szabványnak. Utóbbi fizetési és üzleti gyakorlatokat ír le, amelyek érintik az Egyesült Államok szövetségi visszarúgás-ellenes törvényét (*egyik legfontosabb törvény egészségügyi csalásról és visszaélésről*), de nem minősülnek bűncselekménynek. [16] [17]

Az ingyenes verzió, a Free Plan időkorlát nélkül és korlátlan felhasználóval használható, ám korlátozott az elérhető funkciók terén. A fizetős verzió, a Paid Plan 4 előfizetési opciót kínál (*Basic, Standard, Professional, Enterprise*), amelyek kiegészítik az ingyenes verziót egyéb szolgáltatásokkal, mint pl.: online tárhely, marketing, értékesítés, automatizálás, valamint humán-erőforrás menedzsment. Ezen kívül a fizetős verziókhoz tartozik fejlesztői támogatás és az ingyenesen is elérhető funkciók is részletesebbek. [16] [18]

iDempiere

Nyílt forráskódú szoftver, amely teljesen ingyenes, de egyes szolgáltatók felszámolhatnak költségeket az üzemeltetésért vagy karbantartásért. Az iDempiere az OSGi (*Open Services Gateway initiative*) és az ADempiere együtteséből jött létre 2012-ben. [19]

Az OSGi egy nyílt forráskódú Java keretrendszer moduláris alkalmazások fejlesztésére. Az önálló kód könnyen csomagolható és újrafelhasználható. A csomagokat távolról lehet telepíteni, indítani, leállítani anélkül, hogy a gép újraindítása szükséges lenne. [20]

Az ADempiere (*jelentése: teljesíteni*) szintén nyílt forráskódú ERP és CRM rendszer. 2006-ban jött létre egy hosszan tartó viszontagságos együttműködés után, mely a ComPiere Inc. fejlesztői és a projekt körül kialakuló közösség között volt megfigyelhető. A közösség nehezményezte, hogy a projekt túl nagy hangsúlyt fektet a rendszer kereskedelmi jellegére és nem eleget a tudás megosztására. A közösséget többek között fejlesztők, tesztelők, szakértők, diákok, hatóságok, vállalkozások alkotják, valamint egy kuratóriumi tanács felelős a közösségért és a projekt növekedéséért. [21]

Az iDempiere egyben vállalatirányítás-, ügyfélkapcsolat-kezelő- és ellátásilánc-menedzsment (*Supply Chain Management*) rendszer. Elsősorban a kisvállalkozások számára lett kitalálva. A fejlesztésében, fejlődésében és támogatásában nagy tudású fejlesztők, szakmai szakértők és végfelhasználók egyaránt részt vesznek, így garantálják a magas minőségű szoftvert. Céljuk, hogy növekedésre ösztönözzék a vállalkozásokat és a nyílt forráskód miatt megosztják a tudásukat a fejlődés érdekében. Elérhető webes és asztali alkalmazásként is ugyanazokkal a modulokkal, funkciókkal. Jól skálázható, testreszabható, bővíthető különböző modulokkal és sok külső alkalmazás is integrálható. Hátránya, hogy a szerepkörök változtatásakor kijelentkezés és bejelentkezés szükséges. [19] [22]

Odoo

Az Odoo egy nyílt forráskódú, felhő alapú tartalomkezelő rendszer. Rengeteg alkalmazással támogatja a cégeket az üzleti folyamataikban, kimutatásokat készít, pontos becsléseket ad, valamint növekedésre ösztönzi a vállalatokat. [23]

Története 2005-re nyúlik vissza, amikor Fabien Pinckaers a TinyERP (*jelentése: kis ERP*) fejlesztésébe kezdett. A kezdeti motiváció egy nyílt forráskódú ERP készítése volt, amely nagyobb lesz, mint az SAP. Mivel a cél a növekedés és piacvezetővé válás volt, a TinyERP névválasztás nem volt előnyös, ezért 3 évvel később átnevezték OpenERP-re. [24]

Az OpenERP fejlesztői és a köré épülő közösség folyamatosan nőtt, élvezték a nyílt forráskódból adódó előnyöket. 2010-ben már több, mint 100 alkalmazottal, számos modullal rendelkeztek, de szemléletváltásra volt szükség. Inkább az ügyfelek igényeit

teljesítették, mint hogy egy kiemelkedő szoftvert alkossanak, ezért üzleti modellt váltottak. Szolgáltatásokat nyújtó cégből szoftverkiadó cég lettek. Ehhez támogatókat és befektetőket kerestek, csapatot toboroztak és éveken át tartó növekedésnek indultak. Több, mint 100 országban, 500 partnerrel szerződtek le. A Big Four (*a világ 4 legnagyobb szolgáltató hálózata: Deloitte, Ernst & Young, KPMG és PricewaterhouseCoopers*) elemzői is elkezdtek előnyben részesíteni az OpenERP-t a nagy riválissal, az SAP-val szemben. Nem utolsósorban Franciaországban kötelező érettségi tárgy lett a Word, Excel, PowerPoint mellett. [24] [25]

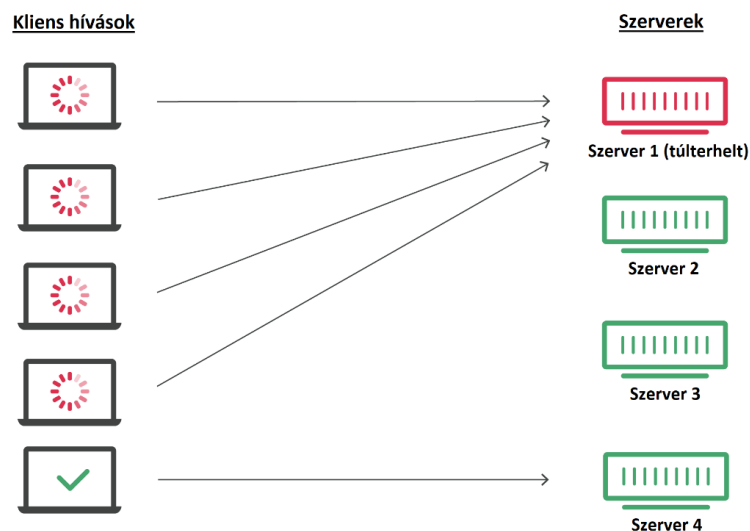
2014-re már több, mint 3000 modullal rendelkezett az eredetileg ERP-nek indult projekt. A hagyományos ERP ágazatokon kívül (*pl.: könyvelés, készletezés, beszerzés, értékesítés vagy humán erőforrás menedzsment*) kiegészült tartalomkezelő rendszerrel (CMS – *Content Management System*), e-kereskedelemmel és üzleti intelligenciával (*Business Intelligence*). Az OpenERP tartalomkezelő rendszere a versenytársaktól is sok elismerést kapott. Az összes weboldal 22%-a Wordpress tartalomkezelővel készült, míg ekkor az OpenERP 0%-on állt. A következő kihívás a marketing fellendítése és új területek lefedése volt. Ehhez újabb támogatókat kerestek és átnevezték a céget és a szoftvert is Odoo-ra. [24]

Az Odoo ingyenes, ha csak egy alkalmazást vesznek igénybe, valamint korlátlan számú felhasználóval lehet használni. A fizetős opciói a közül az alapértelmezett csomag tartalmaz minden Odoo Online alkalmazást, ennek havidíja 11,9 € felhasználónként. Az egyedi csomag szintén tartalmazza az összes Odoo Online alkalmazást, valamint kiegészül például helyileg futtatott Odoo alkalmazásokkal, Odoo Studioval (*alkalmazás építő, amihez nem feltétlen szükséges fejlesztői tudás*) és külső API-val (*Application Programming Interface*), ennek havidíja 17,9 € felhasználónként. [26] [27]

2.3. Terheléselosztás

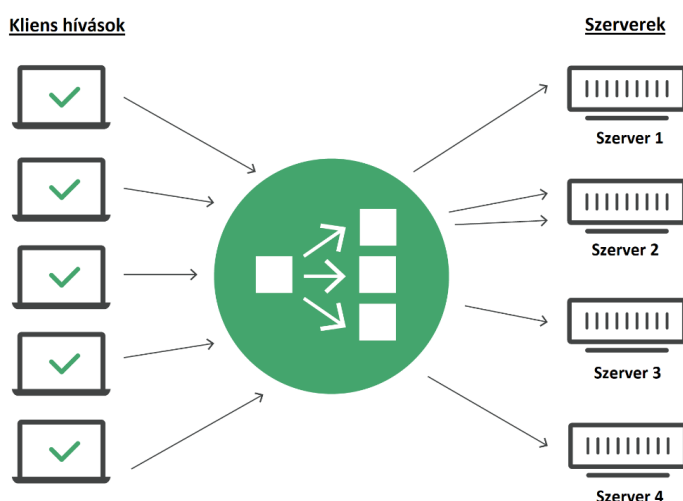
Mivel a cél egy vállalatirányítási információs rendszer készítése kis- és középvállalkozások számára, ezért fel kell készülni arra az esetre is, hogy egyszerre akár 250 fő fogja használni az alkalmazást. Mivel ezek jelentős terhelést jelenthetnek a rendszernek, emiatt optimalizálni kell az adatok bevitelét és olvasását, terheléselosztást kell alkalmazni. Enélkül előfordulhatnak hosszú várakozási idők, szolgáltatás kiesés vagy akár alkalmazás leállás is.

Az 5. ábrán egy példát láthatunk a terheléselosztás nélkül megvalósított kérések kiszolgálására.



5. ábra Kliens-szerver kommunikáció terheléelosztás nélkül [28]

A terheléelosztás a hálózati forgalom és a számítási terhelés több szerver között történő elosztása, ezt szemlélteti a 6. ábra. Csökkenti egyes szerverek terhelését, túlterhelés vagy szerver kiesés esetén egy másik szerver veszi át a feladatát. Növeli a weboldalak elérhetőségét, csökkenti a késleltetést, valamint növeli a teljesítményt. A terheléelosztók folyamatos ellenőrzéseket végeznek, hogy a szerverek elérhetőek-e, valamint mennyire túlterheltek és ez alapján rendeli a kérést az adott szerverhez. Abban az esetben, hogyha egy szerver állapota nem megfelelő, akkor a terheléelosztó eltávolítja a szervert az elérhető szerverek közül. Amint újra ideális állapotú a szerver, visszakerül a szerver készletbe. A folyamatban résztvevő szerverek lehetnek fizikai szerverek és virtualizált szerverek is. Igény esetén a terheléelosztók új virtualizált szervereket is létrehozhatnak. A terheléelosztók lehetnek hardveres és szoftveres megoldások is. A hardveres terheléelosztókhoz külön terheléelosztó eszközök szükségesek. [28] [29]



6. ábra Kliens-szerver kommunikáció terheléelosztással [28]

A terheléselosztókat gyakran használják webalkalmazásokban, hogy kiegyenlítsék az egyes szerverek igénybevételét. Egyes felhőalapú megoldások képesek világszinten egyenletesen tartani a forgalmat a világszerte elhelyezett szerverekkel, ezt hívjuk globális szerverterhelés-elosztásnak (*GLSB – Global Server Loading Balancing*). [28]

A böngésző és a webszerver közötti titkosított kapcsolatot használ. Amikor ezt a terheléselosztó visszafejti a kérés továbbítása előtt, SSL-lezárásnak hívjuk. Ezzel a terheléselosztó erőforrást takarít meg a webszerver számára, így növekszik az alkalmazás teljesítménye. Ez azonban biztonsági kockázatokkal jár, mivel a terheléselosztó és a webszerver közötti adattovábbítás már nincs titkosítva. Amennyiben ezt a módszert szeretnénk használni, akkor érdemes a webszervert és a terheléselosztót ugyanabba az adatközpontba elhelyezni, így csökken a sikeres támadások esélye. A másik módszer, amikor a terheléselosztó egy titkosított kérést továbbít a webszervernek, ezt nevezzük SSL-átengedésnek (*SSL - Secure Sockets Layer*). Ilyenkor a webszerver végzi el a dekódolást, ami teljesítmény csökkenést eredményez, viszont biztonsági szempontból sok vállalat inkább ezt a megoldást választja. [28]

2.3.1. Terheléselosztó algoritmusok

A terheléselosztók különböző algoritmusok alapján határozzák meg a kérések továbbítását. Ezek lehetnek statikus és dinamikus algoritmusok. [28]

A statikus terheléselosztó algoritmusok a rendszer aktuális állapotának figyelembevétele nélkül, egy előre meghatározott terv alapján osztja ki a feladatokat. Ez könnyen beállítható, viszont hátránya, hogy nincs visszajelzés mely szerverek vannak jobban kihasználva, így továbbra is túlterheltek lehetnek a szerverek. A statikus terheléselosztás 2 típusa a körbeforgó (*Round-Robin*) és a kliensoldali véletlenszerű terheléselosztás. [28]

A dinamikus terheléselosztó algoritmusok figyelik a szerverek elérhetőségét, kihasználtságát. A túlterhelt szerverek helyett a kevésbé leterhelt szerverekre irányítják a kéréseket, így a kihasználtság egyenletes marad. Ennek konfigurálása jóval bonyolultabb, mint a statikus terheléselosztás, mivel több tényezőt is figyelembe kell venni, pl.: a szerverek állapotát, teljes kapacitását, valamint a kiosztott feladatok méretét. [28]

Round-Robin algoritmus

A terheléselosztást egy DNS szerver végzi, mégpedig olyan módon, hogy mindig a lista tetején elhelyezkedő szerver szolgálja ki a kérést, majd a szerver a lista végére kerül. A következő kérést az újonnan a lista elejére kerülő kiszolgáló teljesíti. Ebből következik, hogy ez az algoritmus egy statikus terheléselosztási módszer, mivel nem veszi figyelembe a szerverek aktuális állapotát. Akkor használható, ha ugyanazt a tartománynevet több webszerver szolgálja ki. Ilyenkor a kiszolgálók mindegyike különböző IP (*Internet Protocol – egyedi hálózati azonosító*) címet old fel. Akkor a leghatékonyabb, ha azonos specifikációjú kiszolgálókat használ és nincs sok állandó kapcsolat. [29] [30]

Hátránya, hogy nem biztosít egyenletes terheléelosztást a DNS-gyorsítótár (*DNS-cache*) és a kliensoldali gyorsítótár (*client-side cache*) miatt, így nagyobb forgalom alakulhat ki egyes webszervereknél. A szerver leállítását a DNS szerver nem érzékeli, továbbra is megtartja a szerver IP-címét a hurokban. Így, ha van N darab szerver, amiből egy darab nem elérhető, akkor minden N-edik felhasználó kérése nem lesz kiszolgálva. Továbbá a Rond-Robin algoritmus nem veszi figyelembe a kérések bonyolultságát, a földrajzi távolságot és a szerver leterheltségét. Egyes továbbfejlesztett Round-Robin algoritmusok már képesek érzékelni a nem elérhető kiszolgálókat és ezeket kiveszik a hurokból, viszont a gyorsítótár problémára nem adnak nyújtanak megoldást, így a túlterhelés továbbra is előfordulhat. [30]

Legkevesebb kapcsolat módszere

A legkevesebb kapcsolat módszere (*Least connection method*) egy dinamikus terheléelosztó algoritmus, amely figyelembe veszi a szerverek állapotát a kérések kiszolgálóhoz való irányítása során, mégpedig olyan módon, hogy a legkevesebb aktív kapcsolattal rendelkező szerverek kerülnek kiválasztásra. Akkor célszerű használni, ha több hasonló teljesítményű szerver áll rendelkezésre és nagy számú állandó kapcsolat van, amelyek kapcsolódási ideje is eltérő, így egyenlőtlenül oszlik el a terhelés a szerverek között. [29] [31]

A súlyozott legkevesebb kapcsolat módszere a legkevesebb kapcsolat módszeren alapul, valamint kiegészül a kiszolgálók súlyozásával is. A terheléelosztó súlyokat rendel a kiszolgálókhoz az eltérő erőforrásaik és relatív feldolgozási teljesítményük alapján. Ha két szerver rendelkezik a legkevesebb aktív kapcsolattal, akkor a nagyobb súlyozású kiszolgáló lesz kiválasztva. [31]

IP hasítás módszere (IP hash)

Az IP hasítás módszerénél a DTC (*DNS Traffic Control, DNS – Domain Name System - tartománynévrendszer*) a forrás IP címét felhasználva generál egy egyedi hash kulcsot, amely alapján hozzárendeli a klienst a szerverhez. Ez a kulcs újragenerálható, ha a munkamenet megszakad, így biztosítva van, hogy a kliens ugyanarra a kiszolgálóra legyen továbbítva, amelyet korábban használt. Ez a módszer biztosítja a munkamenet folytonosságát, növeli a rugalmasságot és a felhasználói élményt, mivel az újracsatlakozás után ugyanott folytathatja a felhasználó a munkáját, ahol a kapcsolat megszakítása előtt volt. [32] [33]

Erőforrás alapú módszer

Az erőforrás alapú (*vagy adaptív*) módszer esetében a kiszolgálókról lekért állapotjelzők alapján történik meg a továbbítás. Az állapotjelzőket a szervereken futó „ügynök” (*agent*) határozza meg, ezeket kérdezi le minden kiszolgálóról a terheléelosztó, majd beállítja a kiszolgáló súlyozását. Ez a módszer használata előnyös olyan alkalmazásoknál, ahol eltérő terhelések fordulnak elő, mivel részletes terhelés és állapot-ellenőrzés történik. [31]

2.3.2. Hardveres és szoftveres terheléselosztás összehasonlítása

Az alkalmazáskézbésítés-vezérlők (*ADC - Application Delivery Controller*) feladata, hogy biztosítsák az alkalmazások elérhetőségét, garantálják a sebességet a terhelés optimalizálásával, megkövetelik a biztonsági feltételek teljesítését, valamint hiba esetén detektálják azt és a forgalmat átirányítják egy másik kiszolgálóra. A terheléselosztók lehetnek hardver eszközök vagy szoftveres megoldások. Jelentős különbségek vannak, de a közöttük lévő határ kezd eltűnni. [34] [35]

Hardveres terheléselosztás

Hardveres terheléselosztás esetén a speciális processzorokra optimalizálásnak köszönhetően nagy teljesítmény növekedést lehet elérni. Kiegészíthetik alkalmazás-specifikus áramkörökkel, amelyek az adatkezelést gyorsítják fel és közben minimális terhelést kap a központi processzor. A forgalom növekedésekor új terheléselosztó hardverek hozzáadásával dinamikusán lehet növelni a teljesítményt. [34] Ezek a hardverek rack és stack eszközök, ami a hardverek összeszerelésére, elrendezésére utal és figyelembe veszi az üzemeltető helyiség fizikai paramétereit. A függőleges építkezésnek köszönhetően javítja a helykihasználást, valamint biztosítja a megfelelő hűtést és szellőzést. [36] [37]

A hardveres megoldás előnye, hogy jó teljesítményt tud elérni a nagy erőforrás-igényű feladatoknál, mint pl.: SSL-kitöltés, szolgáltatás megtagatással járó támadás (*DDoS – Distributed Denial of Service*) vagy az UDP (*User Datagram Protocol*) forgalom feldolgozása. Magas szintű biztonságot szolgáltat, mivel az eszközhez fizikailag csak a szervezet fér hozzá. Ezek az eszközök rendelkezhetnek olyan biztonsági és tanusítványbeli szabályokkal, amelyek miatt igen elterjedt a kormányzati szektorban. [29] [34] [35]

Hátránya, hogy szakértőkre van szükség a konfiguráláshoz, valamint az eszközökön futó firmware frissítése is bonyolultabb lehet, mint egy operációs rendszer vagy alkalmazás esetében. A firmwarek frissítése szükséges biztonsági és teljesítménybeli szempontból is. A beüzemeléshez szükséges szakértői költségek mellett a fizikai gépek beszerzésével is számolni kell. Hogyha a rendszer eléri a kapcsolatok korlátait, akkor vagy szolgáltatás megtagadásra vagy lassulásra kell számítani. Ennek elkerülése érdekében újabb terheléselosztó berendezések csatlakoztatása szükséges. [29] [35]

Szoftveres terheléselosztás

A szoftveres terheléselosztók lényegében ugyanazokat a funkciókat nyújtják, amelyeket a hardveres terheléselosztók. Csupán néhány funkció hiányzik, mint például az Active Directory (*bizonyos Microsoft hálózati szolgáltatások gyűjtőneve*) és a Kerberos hitelesítési integráció vagy az SSL hardveres gyorsítás. A szoftveres terheléselosztást szokták még alkalmazáskézbésítés-vezérlőnek (*ADC - Application Delivery Controller*)

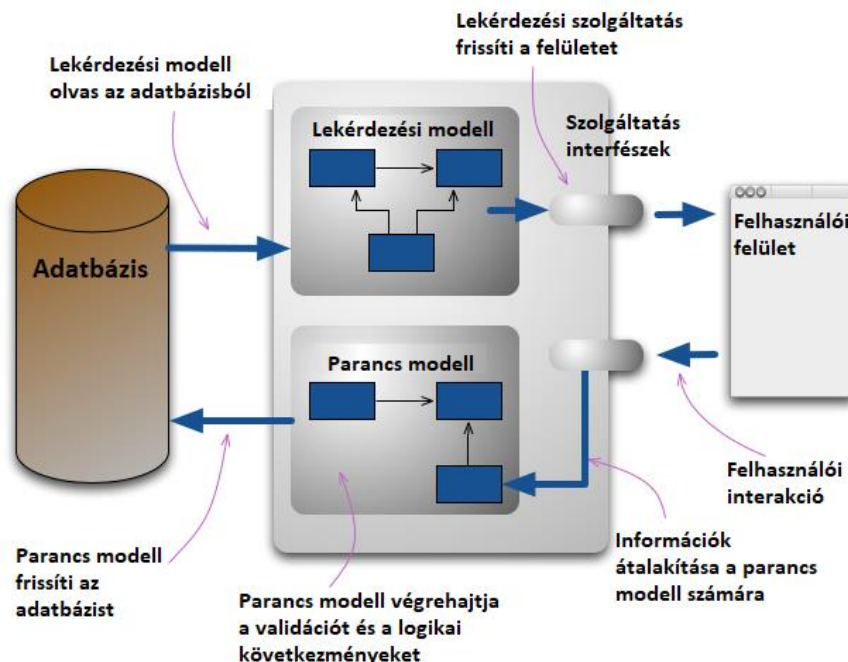
is hívni. Gyakran kész virtuális gépekként valósulnak meg a szoftveres terheléelosztók. [35]

Nagy előnye, hogy olcsóbb, szabványos x86 hardvereken is lehet futtatni. Az üzemeltetése és karbantartása is olcsóbb, mint a fizikai eszközök esetében. Rugalmasan skálázható az igényeknek megfelelően. Ha a forgalom eléri a kitűzött határt, akkor további szoftver példányok hozzáadásával lehet növelni a teljesítményt. Lehetővé teszi a felhő alapú megoldás, valamint a hibrid megoldást is, pl.: a fő terheléelosztó helyben van megvalósítva, a tartalék terheléelosztó pedig felhőben. [29]

Hátránya, hogy a frissítések beszerzése költségekkel járhat, valamint a terheléelosztó érzékeny lehet az operációs rendszer verziókra és a virtuális gépeket futtató hipervizor módosításokra, frissítésekre. Az ebből adódó hibák kiküszöböléséhez speciális tesztelésre van szükség. [29] [35]

2.3.3. Parancs-lekérdezés felelősség elkülönítése

A terheléelosztás mellett fontos a nagy erőforrás-igényű feladatok gyorsítása, optimalizálása is. A parancs-lekérdezés felelősség elkülönítése (*CQRS - Command Query Responsibility Segregation*) egy tervezési minta, melynek folyamatát a 7. ábra szemlélteti. Lényege, hogy az adatok frissítéséhez és olvasásához más modellt használ. Ezen kívül a parancskezelők felelősek a nagy erőforrás-igényű lekérdezések leegyszerűsítéséért. A minta kifejlesztése Bertrand Meyer nevéhez fűződik, aki megfogalmazta, hogy egy eljárás vagy az adatok lekérdezésével foglalkozhat, vagy a módosítással, de mindkettővel nem. [38]

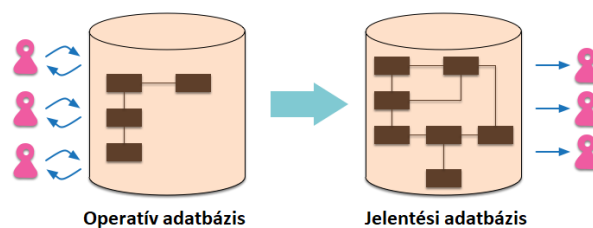


7. ábra Parancs-lekérdezés felelősség elkülönítése [39]

Nagy általánosságban az információs rendszerekkel történő interakciókat CRUD (*Create* - létrehozás, *Read* - olvasás, *Update* - frissítés, *Delete* - törlés) adattárként használják. Adott egy modell, amely leírja a szerkezetet. Új rekordokat hozhatunk létre, módosíthatjuk, törölhetjük őket, majd ezeket tároljuk és olvassuk. Ahogy egy rendszer növekszik, az alap modelltól egyre távolabb kerül és létrejönnek új rekord típusok, esetleg virtuális rekordok. A virtuális rekordok lehetőséget adnak a fizikailag tárolt adatok ábrázolására az igényeknek megfelelő formában, pl.: rekordok összezsukása, csoportosítása. [39]

A CQRS legtöbbször a parancs tervezési mintával van megvalósítva, amely a kérést egy olyan objektummá alakítja, amely tartalmazza a kéréssel kapcsolatban az összes információt. Így a rendszernek lehetősége lesz a kéréseket metódus argumentumként átadni. Ennek köszönhetően a kérések egy későbbi időpontban reprodukálhatóak, végrehajthatóak egy másik eszközön, adatbázison is. A kezelők (*handlers*) végrehajtják ezeket a lépéseket, közben naplózhatják az eseményeket, hibákat, valamint hiba esetén visszaállítást végeznek. A tranzakciók egymás utáni végrehajtásakor az aktuális tranzakció eredményeként létrejön egy esemény, amely meghatározza a következő lépést. Amennyiben egy lépés hibára fut, akkor a kompenzáló tranzakció kezeli az előző tranzakciókat. Ezt hívjuk *saga* mintának. [38] [40] [41]

A CQRS bevezetése sokkal bonyolultabbá teszi a rendszert és növeli a hibalehetőséget is, mivel az egyszerű SQL műveletek helyett parancskezelők szolgálják ki a kéréseket. Ez kiegészítő kódot igényel, szükség esetén a már működő rendszereket ki kell egészíteni a megfelelő részletességű naplózással is. Az íráshoz és az olvasáshoz használt modell általában elegendő átfedést tartalmaz, de jelentősen eltérhetnek egymástól. Hogyha a parancskezelők bevezetése túl nagy bonyolultsággal járna, vagy nincs szükség rá, akkor lehetőség van csak a lekérdezéseket optimalizálni. Ehhez lehet redundáns táblákat használni, amelyeket ajánlott külön adatbázisban tárolni, úgynevezett jelentési adatbázisban (*reporting database*). Az elkülönített jelentési adatbázist a 8. ábra mutatja be. A jelentési adatbázis szerkezete illeszkedik a lekérdezésekhez. Nagy előnye, hogy nem szükséges a normalizálás, mert a felhasználók oldaláról csak olvasható. A jelentési adatbázison futtatott lekérdezések nem befolyásolják az operatív adatbázis terhelését. Hátránya, hogy naprakészen kell tartani az adatokat, mely megköveteli, hogy az adatváltozások kommunikálva legyenek. Igény esetén ez a kommunikáció azonnal történhet, de gyakoribb, hogy éjszakai futtatással oldják meg. [38] [40] [42]

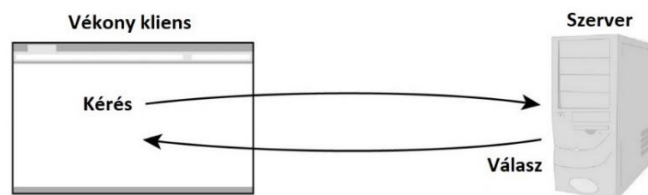


8. ábra Jelentési adatbázis [42]

3. TERVEZÉS

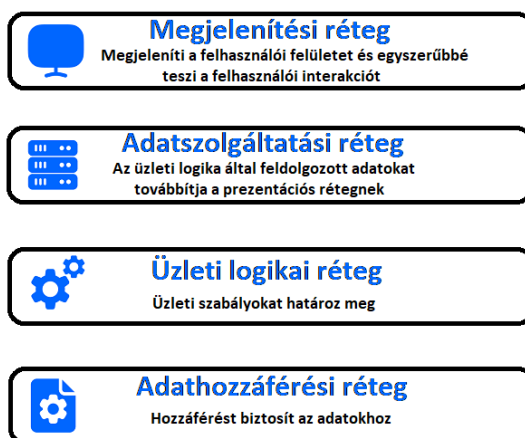
A következő fejezetben egy vállalatirányítási információs rendszer tervezését fogom bemutatni. A cél egy olyan ERP készítése, amely képes kiszolgálni a kis- és középvállalkozások igényeit és segít áttekinteni a humán erőforrás és projekt költségeket, valamint rögzíti a napi tevékenységeket. Mindez úgy lesz megvalósítva, hogy csak autentikáció után és csak a megfelelő authorizációval rendelkező felhasználók férjenek hozzá az alkalmazás adott területeihez.

A felhasználók böngészőn keresztül, vékony klienset használva érik el a funkciókat, ennek folyamatát a 9. ábrán láthatjuk. Így a használatához nincs szükségük sok erőforrásra, mert a kéréseket egy távoli gép fogja kiszolgálni.



9. ábra Kérések kiszolgálása vékony kliens használatával [43]

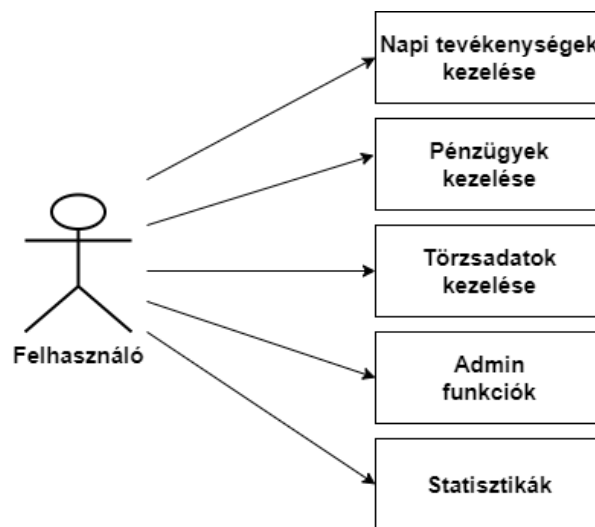
A webalkalmazás naprakész, platformfüggetlen technológiával lesz elkészítve, alkalmazva a rétegelt alkalmazás jellemzőit, mely a 10. ábrán látható. Az adathozzáférési réteg biztosítja az adatbázisban tárolt adatok létrehozását, olvasását, módosítását és törlését. Az üzleti logikai réteg biztosítja, hogy az adatok csak a megfelelő validáció után kerüljenek be az adatbázisba, valamint helytelen adat esetén hibát dob, ami a felületre kivezelve visszajelzést ad a felhasználónak. Az adatszolgáltatási réteg biztosítja a CRUD műveleteket, míg végül a megjelenítési réteg felel a felhasználói felületért és a felhasználói interakciók kezeléséért. Itt cél a nagy számú adatbevitel megkönnyítése, a nap tevékenységeknél a csoportos hozzáadás lehetősége.



10. ábra Rétegelt alkalmazás szintjei [44]

Az adatok adatbázisban lesznek tárolva. Egy előre létrehozott adminisztrátorral (*superadmin*) bejelentkezve létre lehet hozni adminisztrátort az ügyfél részére, akinek lehetősége lesz a vállalkozás alkalmazottait hozzáadni és szerepköreiket beállítani. Bizonyos törzsadatok, mint például az országok vagy felhasználói szerepkörök az alkalmazás első indításakor generálva lesznek. Mivel a napi szintű tevékenységek rögzítése nagy számú adatot fog eredményezni, ezért szükséges a lekérdezéseket optimalizálni, valamint a nagy erőforrás-igényű lekérdezéseket előre kiszámítani. Erről részletesebben a kimutatások készítése fejezetben fogok írni.

Az alkalmazás 5 főbb területre lesz osztható, amelyek a 11. ábrán láthatóak. Ezek a napi tevékenységek, pénzügyek, törzsadatok kezelése, admin funkciók és statisztikák áttekintése.



11. ábra Alkalmazás funkciók áttekintése

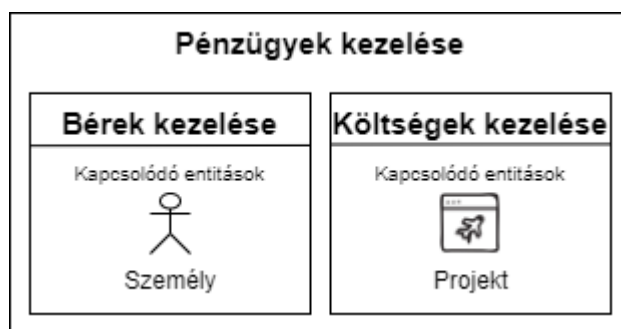
A napi tevékenységek kezelése a napi munkaidő beosztásban, szabadságok és betegszabadságok kezelésében nyilvánul meg. A napi munkaidő beosztásnál meg lehet adni, hogy munkaszüneti napról van-e szó, mely projekten dolgozik egy vagy több személy, hány órát vett igénybe a munka, valamint mennyi idő volt szükséges az utazáshoz. Hogyha autóval utazik a helyszínre, akkor a listából ki lehet választani az autót is. A szabadságok és betegszabadságok kezelése is ebben a menüpontban történik. Amennyiben egy személy egy adott napon már hozzá van rendelve egy projekthez, akkor nem lehet szabadsághoz, betegszabadsághoz hozzáadni. A szabadsághoz, betegszabadsághoz hozzáadáskor is történik validáció, hogy az adott személy ne legyen még hozzárendelve projekthez aznapra. Ezen kívül a belsős és külsős dolgozók külön kerülnek listázásra.

A napi tevékenységek kezeléséről és a folyamatban érintett entitásokról a 12. ábra ad áttekintést.



12. ábra Napi tevékenységek áttekintése

A pénzügyek kezelése a két pontból áll, amint azt a 13. ábra szemlélteti. Ezek a bérek és a költségek kezelése. A béreknél a személy kiválasztása után meg lehet adni havi bért, napidíjat vagy órabért attól függően, hogy milyen formában van foglalkoztatva a dolgozó. A bér érvénybe lépéséhez tartozik egy dátum is, ezt és az értéket felhasználva statisztika készül a bér alakulásáról, amely grafikus elemekkel lesz megjelenítve. A költségek menüpontnál ki lehet választani, hogy melyik projekt költségeit szeretnénk áttekinteni, valamint új költséget is hozzáadhatunk.



13. ábra Pénzügyek áttekintése

A törzsadatok alá tartoznak a személyek, a cégek, a projektek, az autók, a beosztások, valamint a címek kezelése. A személyek esetén a személyes adatok, a beosztás, a szabadságok és a betegszabadságok lesznek tárolva. A cégek neve mellett a címüket és adószámukat lehet megadni. A projekteknél az azonosító, a megbízó ügyfél, a bevétel és a tervezett költség, valamint kezdő dátum és határidő kerül rögzítésre. Az autók esetén márkájuk, típusuk, valamint rendszámuk lesz tárolva.

A törzsadatok alá tartozó területeket és a hozzájuk kapcsolódó entitásokat a 14. ábrán láthatjuk.



14. ábra Törzsadatok áttekintése

A statisztikák frissítése, a felhasználók hozzáadása és a szerepkörök felhasználókhoz rendelése az admin funkciókhoz tartoznak, melyről a 15. ábra ad áttekintést. Ezt a menüpontot csak az adminisztrátorok fogják elérni.



15. ábra Admin funkciók áttekintése

Utolsó területként a statisztikákat említeném meg, melyek segítenek áttekinteni a projektekből származó bevételek és a rájuk könyvelt munkaórák költségeit és az egyéb projekt költségeket. Fontos, hogy ezek a lekérdezések rövid válaszidővel adjanak eredményt, még akkor is, ha nagy adatmennyiséget kell feldolgozniuk. Ehhez redundáns táblákat fogok használni, amelyek az adott lekérdezésekhez igazodnak. Az így keletkezett adatok sok esetben átfedést mutatnak, de a teljesítmény növelése érdekében ez egy elfogadható hátrány. Sok statisztikai adathoz nem szükséges a legfrissebb adatokat használni, ezért a redundáns táblák adatainak frissítése történhet automatikusan néhány óránként, vagy olyan időpontban, amikor a rendszert nem- vagy kevesen használják, például éjszaka. Abban az esetben, hogyha fontos, hogy a lekérdezés a legfrissebb adatokkal dolgozzon, akkor a frissítés kézzel is indítható.

A hibás vagy nem szükséges adatok törlése nem fizikai törlésként fog megvalósulni, hanem logikai törlésként. Ez biztosítani fogja, hogy nem lesz véglegesen -akár véletlenül- adat törölve, a törölt adatok visszaállíthatóak. Mindegyik entitáson lesz egy mező, ami a törlés dátumát mutatja. Ez a mező alapértelmezetten NULL (üres) lesz, ha pedig törlik, akkor az aktuális dátum íródik bele.

Fontos szempont az is, hogy az alkalmazás lehetőséget teremtsen a bővítésre, továbbfejleszthetőségre, valamint felhasználóbarát legyen. Képes legyen az összes entitás típus listázására. Ahol lehetséges a csoportos hozzáadás, ott legyen lehetőség több elem kijelölésére. Az alkalmazásba bejelentkezés után a kezdőlapon hasznos információk lesznek, mint pl.: kik vannak az adott napon szabadságon és betegszabadságon, mely dolgozók nincsenek még beosztva egy projektre sem, valamint itt lehet majd áttekinteni a projektekre könyvelt munkaórákat, költségeket a bevételhez viszonyítva.

A felhasználók csak azokat a menüpontokat fogják látni, amelyekhez van jogosultságuk. Ha közvetlenül az adott címet (*URL – Uniform Resource Locator*) használnák az eléréshez, akkor szintén csak a szükséges szerepkör megléte esetén lesz jogosultságuk az oldal megtekintéséhez.

Amennyiben egy felhasználó bejelentkezve marad és nem használja az alkalmazást, akkor egy megadott idő után a rendszer automatikusan kilépteti, így csökkentve a visszaélések lehetőségét.

4. A MEGOLDÁSI MÓDSZER KIVÁLASZTÁSA, A VÁLASZTÁS INDOKLÁSA

Mivel a cél egy platformfüggetlen webalkalmazás elkészítése, ezért a kiválasztott technológia ASP.NET Core lesz. Ezzel biztosítva lesznek a vékonykliens előnyei és megvalósítható a rétegelt alkalmazás architektúra. A felhasználók böngészőn keresztül érik el az alkalmazást. Kliens oldalon néhány validáció és a felhasználói élményt növelő funkciók lesznek, míg az üzleti logika nagy része szerver oldalon lesz megvalósítva. A fejlesztéshez Microsoft Visual Studio Community 2019 verziót fogok használni.

Adatbázisnak a Microsoft SQL Server Express lesz kiválasztva. Fejlesztéshez használható az SQL Developer verzió is, de az Express verziót használva szimulálható az ügyfél oldali éles működés is, mivel ez a verzió kerülne kiválasztásra egy élesítés esetén. Az Express verziónak vannak korlátai, mint például az adatbázis csak 1 GB memóriát és csak egy processzor magot használhat, az adatbázis mérete maximum 10 GB lehet, és a puffer mérete is maximum 1 MB lehet. Nagy előnye viszont, hogy ingyenes fejlesztői és kereskedelmi célra is. [45] [46] Az adatbázis táblákból történő kézi lekérdezéshez SQL Server Management Studio alkalmazást fogok használni, mint fejlesztői segédeszközt.

A nagy adatmennyiségből történő lekérdezéseknél elvárás, hogy rövid válaszidővel kapjon a felhasználó eredményt, így a CQRS-ben használt módszerek, mint az eltérő modell és a redundáns adattárolás lesznek használva.

A statisztikai kimutatásokhoz és néhány felületnél, ahol a vizuális áttekintés növeli a felhasználói élményt és az áttekinthetőséget, ott Google Chart lesz a kiválasztott szolgáltatás.

A felhasználói élmény növeléséhez ezen kívül JQuery Javascript könyvtár fog hozzájárulni, amelyet a többszörös kijelöléshez és validációkhoz fogok használni. A szebb megjelenítéshez Bootstrap CSS (*Cascading Style Sheets – stílusleíró nyelv*) keretrendszert és Font Awesome ikon eszközkészletet fogok használni.

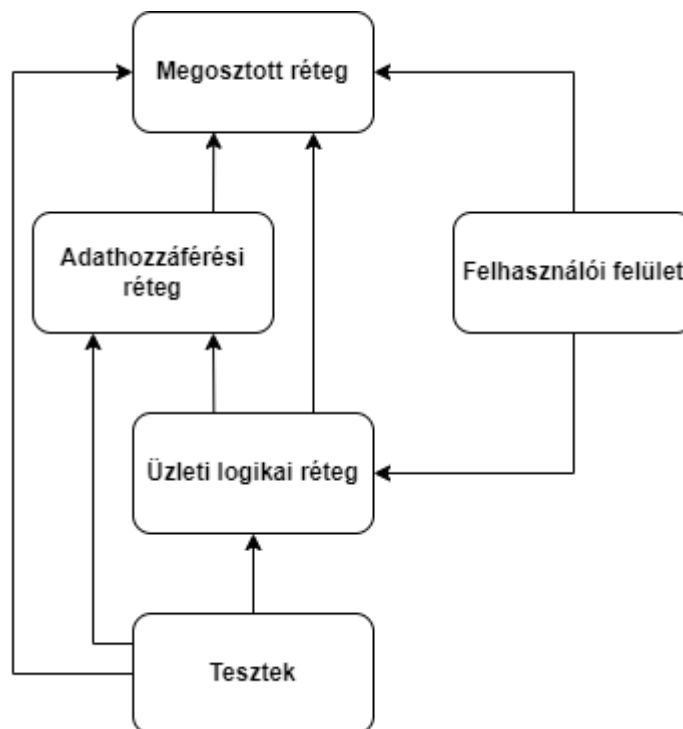
A verziókezeléshez Git-et és SourceTreet fogok használni. A nagyobb egységek külön ágon (*branch*) lesznek lefejlesztve, majd vissza visszamergelve a fő master ágba.

5. MEGVALÓSÍTÁS

A megvalósítás első lépéseként a rétegelt alkalmazás szerkezetét építettem fel. Ezután véglegesítésre kerültek az entitások és a hozzájuk szükséges interfészek, amelyekből az alkalmazás első indulásakor adatbázis táblák jöttek létre. A következő lépés az adathozzáférési réteg megvalósítása volt, majd az üzleti logikai réteg. Ezután kezdődött meg a felületek elkészítése és a funkciók részletes kidolgozása.

5.1. Rétegelt alkalmazás felépítése

A rétegelt alkalmazás felépítése és a további fejlesztések is Visual Studioban történtek. A Solution (*projekteket tartalmazó fájl*) fájlhoz hozzá lett adva a Shared (*megosztott*) réteg, amely entitásokat, interfészeket tartalmaz és az összes többi réteg használja. Ezután következett a Repository (*adathozzáférési*) réteg, amely tartalmazza a Shared projekt referenciát. A következő réteg a BusinessLogic (*üzleti logika*) volt, amely hivatkozik a Repository és a Shared referenciákra. Itt vannak megvalósítva a CRUD műveletek és az egyéb adatmódosításokat végrehajtó műveletek. A UI (*User Interface*) réteg a megjelenítési réteg, amely használja az BusinessLogic és a Shared projekt referenciákat. Itt vannak megvalósítva a felületek és innen történik az üzleti logika metódusainak hívása is. A projekt referenciákat a 16. ábra ábra szemlélteti.



16. ábra Alkalmazás rétegek referenciái

5.2. Interfészek, entitások definiálása

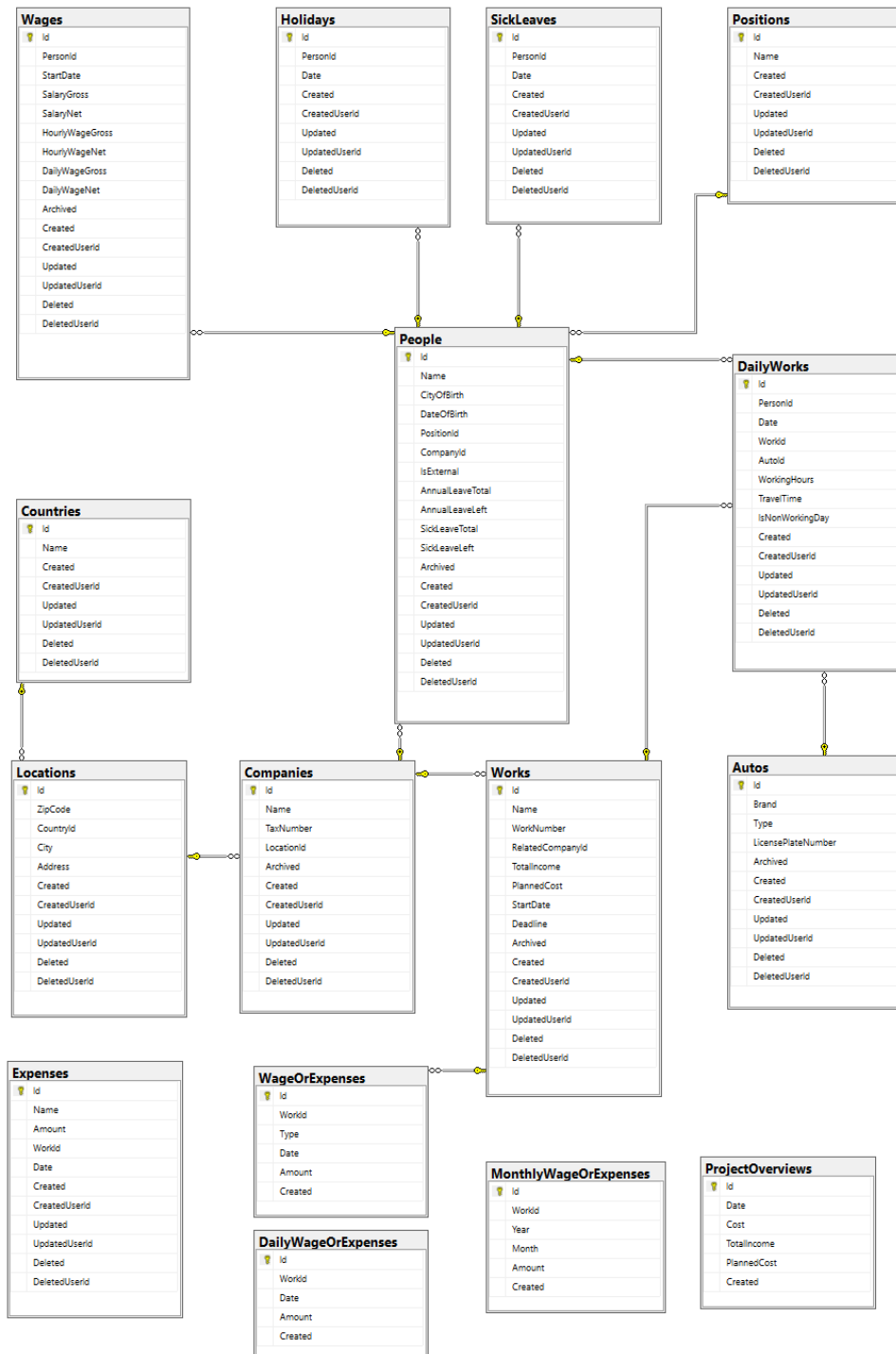
Az első interfész az IDbEntity, amely technikai mezőket ír elő. Ezek az Id (*azonosító*), Created (*rögzítés dátuma*), CreatedUserId (*rögzítő felhasználó azonosítója*), Updated (*módosítás dátuma*), UpdatedUserId (*módosító felhasználó azonosítója*), Deleted (*törlés dátuma*), valamint DeletedUserId (*törlő felhasználó azonosítója*). A létrehozáshoz tartozó dátum és rögzítő felhasználó kötelező mezők, ez minden rekordnál töltve lesz. A módosítás és törlés mezői opcionálisak, ezek akkor lesznek töltve, ha egy rekord módosítva vagy törölve lesz. Ebből következik, hogy a törlés nem fizikai törlést jelent, hanem csak logikai törlést. Azok az entitások, amelyeknél a törlés dátuma ki van töltve nem kerülnek listázásra és a kimutatások sem veszik figyelembe.

A másik fontos interfész az IArchivable, amely egy mezőt tartalmaz, az Archived (*archiválás dátuma*) mezőt. Ezt az interfészt azok az entitások valósítják meg, amelyeknél nem szeretnénk a törlést alkalmazni, viszont valamilyen okból -például egy munkavállaló munkaviszonya megszűnik- már nincs szükség a listázására, de a kimutatások figyelembe kell, hogy vegyék.

Mindegyik entitás a DbEntity entitásból van leszármaztatva, amely megvalósítja az IDbEntity interfészt. Az autó, cég, személy és projekt entitások pedig az IArchivable interfészt is megvalósítják, így lehetőség van ezen entitások archiválására.

5.3. Létrejött adatbázis táblák

Az adatbázis táblákról készült diagram a 17. ábrán látható. A diagram SQL Server Management Management Studio alkalmazással lett generálva.

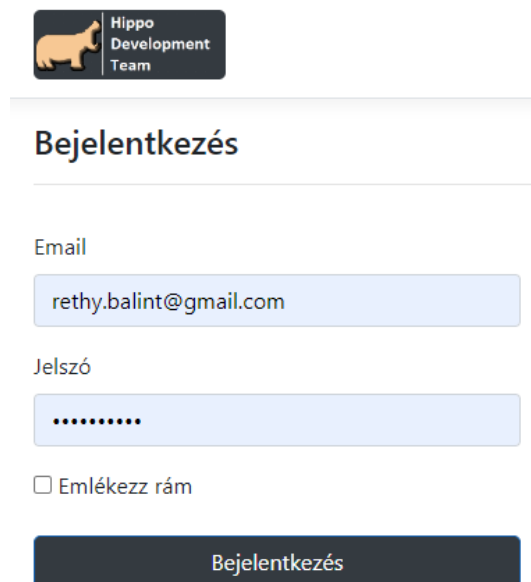


17. ábra Adatbázis struktúra

5.4. Felületek és funkciók megvalósítása

Az alkalmazás első indulásakor létrejönnek az adatbázis táblák és feltöltésre kerülnek bizonyos táblák alapértelmezett adatokkal. Ilyen adatok például az országok, valamint egy superuser (*fejlesztői adminisztrátor*) felhasználó. Ezzel az adminisztrátor fiókkal történő belépés esetén lehetőség van további felhasználókat hozzáadni, akikhez hozzárendelhetők szerepkörök. Célszerű egy ügyfél oldali adminisztrátort hozzáadni, akinek bejelentkezés után lehetősége lesz további felhasználókat létrehozni.

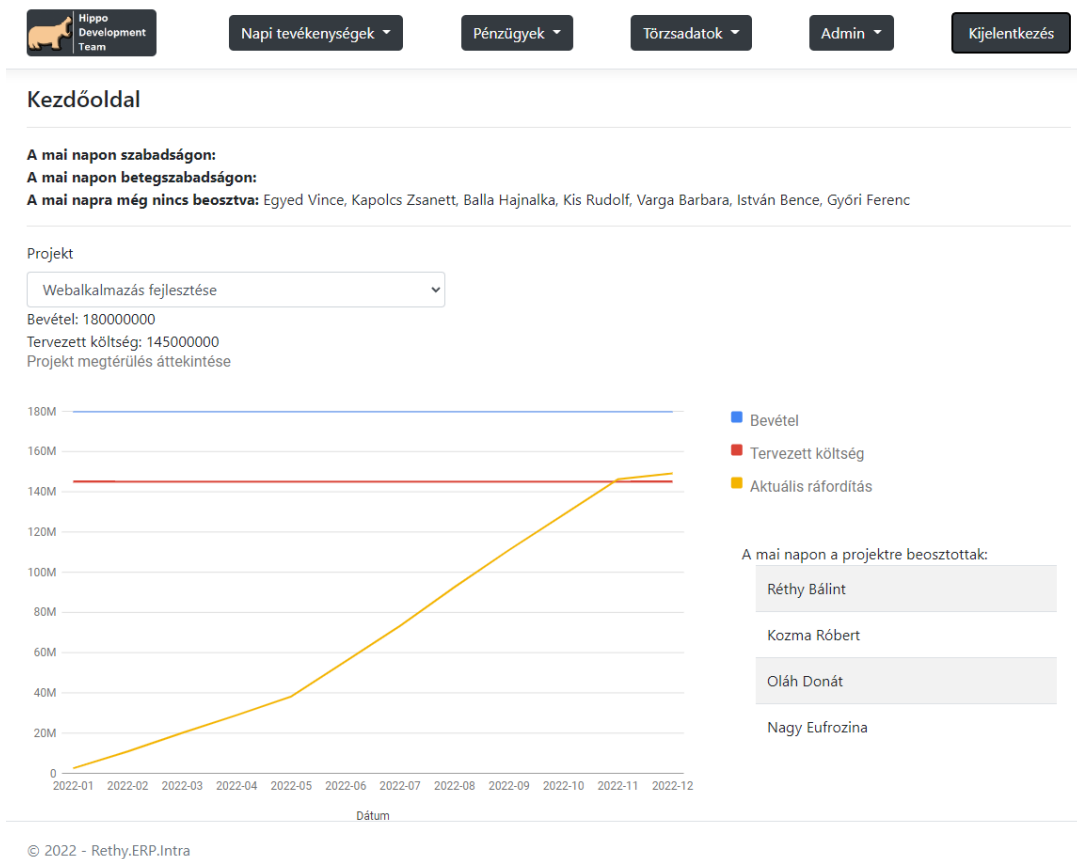
Az alkalmazás összes felületéhez bejelentkezés szükséges. Amennyiben a felhasználó nincs bejelentkezve, akkor a főoldal vagy bármely más oldal megnyitásakor a bejelentkezési oldal nyílik meg, URL-ben pedig tárolva lesz a visszatérési URL, ahova bejelentkezés után át lesz irányítva a felhasználó. A bejelentkezési felület a 18. ábrán látható.



The image shows a login form titled "Bejelentkezés". At the top left is a logo for "Hippo Development Team" featuring a stylized orange hippo. Below the title, there are two input fields: "Email" with the value "rethy.balint@gmail.com" and "Jelszó" (Password) with a masked input of dots. Below the password field is a checkbox labeled "Emlékezz rám" (Remember me). At the bottom is a dark blue button labeled "Bejelentkezés" (Login).

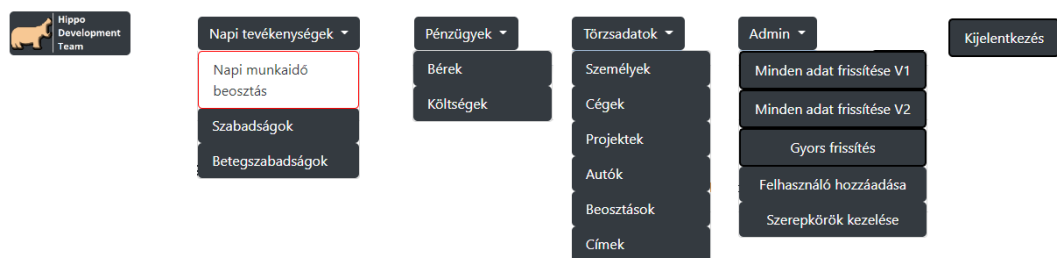
18. ábra Bejelentkezési felület

Bejelentkezés után a főoldal jelenik meg, melyet a 19. ábrán láthatunk. Itt hasznos információk jelennek meg az aktuális napi tevékenységeket illetően. Ilyenek az adott napon szabadságon, betegszabadságon lévő személyek listázása, valamint azok a dolgozók is megjelennek egy külön listában, akik még nem lettek projekthez hozzárendelve. Lehetőség van projekt kiválasztására is, amely egy vonal diagramot rajzol ki a projekt áttekintésére. A projekt bevétele, a tervezett költsége, valamint az aktuális ráfordítások jelennek meg havi bontásban. Ezen kívül az adott napon a kiválasztott projekten dolgozó személyek is megjelennek a diagram mellett egy listában.



19. ábra Főoldal

Az alkalmazás menüpontjai csak a szükséges szerepkör megléte esetén jelennek meg és URL-ből történő hívás esetén is történik autorizációs ellenőrzés. A 20. ábrán az alkalmazásban elérhető menürendszer látható bejelentkezett adminisztrátor szerepkör esetén.

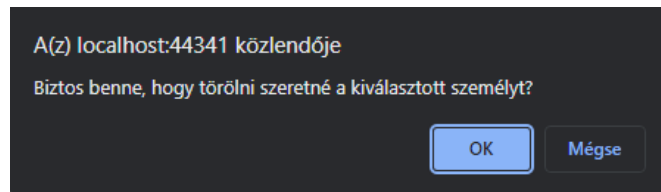


20. ábra Menüpontok

5.4.1. Törzsadatok

A törzsadatok menüpont alá a rendszer alapját képező entitások tartoznak. Ezek a személyek, a cégek, a projektek, az autók, a beosztások, valamint a címek. A törzsadatok listázásához minden esetben adatátviteli objektumokat (*DTO – Data Transfer Object*) használunk, melyek a megjelenítéshez szükséges mezőkön kívül csak egy azonosítót

tartalmaznak, így csökkentve az adatközvetítést. Mindegyik sor jobb szélén megjelennek a megtekintés, szerkesztés és törlés linkek, melyekhez Font Awesome ikonok lettek felhasználva. Megtekintéskor a mezők csak olvashatóak, míg szerkesztéskor módosíthatóak. Az elemek törlésekor egy JavaScript megerősítő ablak ugrik fel, amely csak jóváhagyás esetén folytatja a műveletet. A 21. ábrán egy személy törlésekor felugró megerősítő ablak látható.



21. ábra Törlés megerősítése

Mindegyik DTO-kat tartalmazó ViewModel (*nézet modell*) a BaseViewModel-ből (*saját őssztály*) van származtatva, így mindegyik osztály tartalmazza a CanEdit (*szerkesztés joga*) és CanDelete (*törlés joga*) mezőket. Ezek alapján jelennek meg a listázó felületen a szerkesztés és törlés ikonok.

Személyek

A személyek menüpontra kattintva megjelenik az összes hozzáadott személy, aki nem lett archiválással kivéve az aktív dolgozók közül. Erre a 22. ábrán láthatunk példát.

Személyek

Új személy	
Név	
Réthy Bálint	  
Kozma Róbert	  
Egyed Vince	  
Oláh Donát	  
Kapolcs Zsanett	  
Balla Hajnalka	  
Kis Rudolf	  
Nagy Eufrozina	  
Varga Barbara	  
István Bence	  
Győri Ferenc	  

22. ábra Személyek listázó oldala

Új személy hozzáadására a 23. ábrán láthatunk egy példát. A szabadságokat és betegszabadságokat tartalmazó mezőkön kívül minden mező töltése kötelező. A kötelező

mezők ellenőrzése mellett egyéb validációk is történnek az üzleti logikai rétegben. A születési dátum nem lehet nagyobb az aktuális dátumnál és az év sem lehet kisebb 1900-nál, valamint az összes és a rendelkezésre álló szabadság és betegszabadság nem lehet kisebb, mint 0.

Új személy

Név

Frank Drebin

Születési hely

Florida

Születési idő

1960. 05. 15.

Beosztás

Projektmenedzser

Munkaadó

Hippo Development Team

☒ Külsős dolgozó

Éves szabadság

35

Hátralevő szabadnapok száma

32

Összes betegszabadság

15

Igénybe vehető betegszabadságok száma

0

Hozzáadás

Vissza

23. ábra Új személy hozzáadása

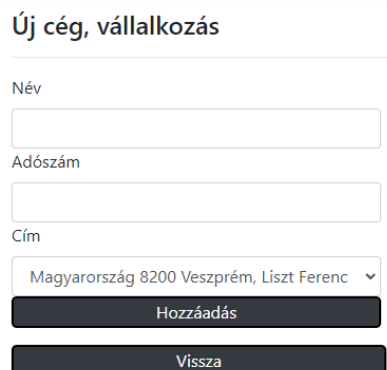
Cégek

Cégek listázásakor csak a cégek neve jelenik meg, a többi mező megtekintés vagy szerkesztés után látható. A cégeket listázó oldalt a 24. ábrán láthatjuk.

Cégek, vállalkozások	
Új cég	
Név	
Hippo Development Team	 
MÁV	 
McDonald's Kistarcsa	 
Spider Club	 
Óbudai Egyetem NIK	 
Hangvilla	 

24. ábra Cégek listázó oldala

Új cég hozzáadásakor a név töltése kötelező, az adószámot később is meg lehet adni. A címet egy legördülő listából lehet kiválasztani, melynek forrása a már hozzáadott címekből áll. A felület megvalósítását a 25. ábra szemlélteti.



Új cég, vállalkozás

Név

Adószám

Cím

Magyarország 8200 Veszprém, Liszt Ferenc

Hozzáadás

Vissza

25. ábra Új cég hozzáadása

Projektek

Projektek listázására a 26. ábrán láthatunk példát. Csak a projekt neve jelenik meg, további információkat a megtekintésre vagy szerkesztésre kattintva érünk el.



Projektek	
Új projekt	
Név	
Hippo Development Team saját költségek	  
Informatikai fejlesztés	  
Webalkalmazás fejlesztése	  
Tanácsadás	  
Alkalmazás támogatás	  

26. ábra Projektek listázó oldala

Új projekt hozzáadásakor a név, a megbízó ügyfél, a kezdő dátum és a határidő megadása kötelező. A munkaszám/azonosító, a bevétel és a tervezett költség opcionális mezők, de amint rendelkezésre állnak az információk ajánlott megadni. A kötelező mezők vizsgálatán kívül ellenőrzésre kerül az üzleti logikai rétegben, hogy a bevétel és a tervezett költség nem lehet negatív szám és a dátum mezők sem helytelenek. Egy új projekt hozzáadására a 27. ábrán láthatunk példát.

Új projekt

Név

Informatikai tanácsadás

Munkaszám/Azonosító

W001

Megbízó ügyfél

MÁV

Bevétel

1000000

Tervezett költség

250000

Kezdő dátum

2023. 01. 03.

Határidő

2023. 01. 27.

Hozzáadás

Vissza

27. ábra Új projekt hozzáadása

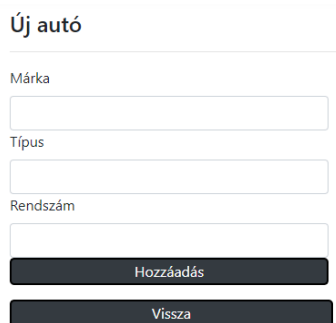
Autók

Az autó listázó felületen megjelenik a márka, típus és a rendszám. Az alkalmazáshoz hozzáadott autók listázó felülete a 28. ábrán látható.

Autók			
Új autó			
Márka	Típus	Rendszám	
Suzuki	Swift	NUK-680	 
Toyota	Corolla	SXQ-015	 
Ford	Mondeo	GWP-090	 
Skoda	Fabia	AZK-484	 
Kia	Ceed	DUJ-064	 
Dacia	Logan	QEY-601	 
Volkswagen	Passat	PHV-088	 
Fiat	Panda	SBX-164	 

28. ábra Autók listázó felülete

Új autó hozzáadásakor mindegyik mező töltése kötelező. A felület megvalósítását a 29. ábra szemlélteti.



Új autó

Márka

Típus

Rendszám

Hozzáadás

Vissza

29. ábra Új autó hozzáadása

Beosztások

A beosztás entitás csak egy megnevezés mezőt tartalmaz. A listázó felületről hiányzik a megtekintés funkció, mivel az nem szolgálna plusz információval. A listázó felületet a 30. ábra mutatja be.



Beosztások

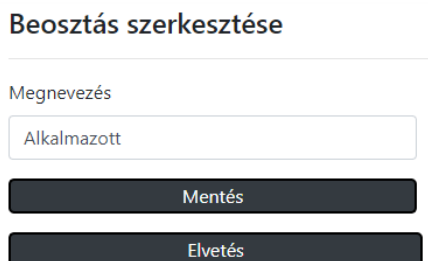
Új beosztás

Név

Alkalmazott	 
Vállalkozó	 
Humán erőforrás	 
Ügyvezető	 
Részlegvezető	 
Projektmenedzser	 
Könyvelő	 

30. ábra Beosztások listázó oldala

Új elem hozzáadásakor és módosításkor a megnevezés mező töltése kötelező. Beosztás módosítására példát a 31. ábrán láthatunk.



Beosztás szerkesztése

Megnevezés

Alkalmazott

Mentés

Elvetés

31. ábra Beosztás módosítása

Címek

A címek listázásakor megjelenített mezők az ország, irányítószám, város, és cím. Az autókhoz hasonlóan a megtekintés nem szolgálna plusz információval, ezért a

megtekintés funkció felesleges. A rendszerhez hozzáadott címek listázó felületét a 32. ábra mutatja be.

Címek

Új helyszín				
Ország	Irányítószám	Város	Cím	Műveletek
Magyarország	8200	Veszprém	Liszt Ferenc u. 1.	 
Magyarország	1087	Budapest	Könyves Kálmán krt. 54.	 
Magyarország	2143	Kistarcsa	Mester u. 5.	 
Magyarország	1036	Budapest	Fényes Adolf u.	 
Magyarország	1034	Budapest	Bécsi út 96/B	 
Magyarország	8200	Veszprém	Bruszniai Árpád u. 2.	 

32. ábra Címek listázó oldala

Új cím hozzáadásakor minden mező töltése kötelező. A kötelező mezők ellenőrzésén kívül vizsgálva van, hogy ha az irányítószám szám, akkor nem lehet negatív. Új cím hozzáadására példát a 33. ábrán láthatunk.

Új cím

Ország

Magyarország

Irányítószám

1077

Város

Budapest

Cím

Wesselényi utca 9. 4. em/1.

Hozzáadás

Vissza

33. ábra Új cím hozzáadása

5.4.2. Pénzügyek

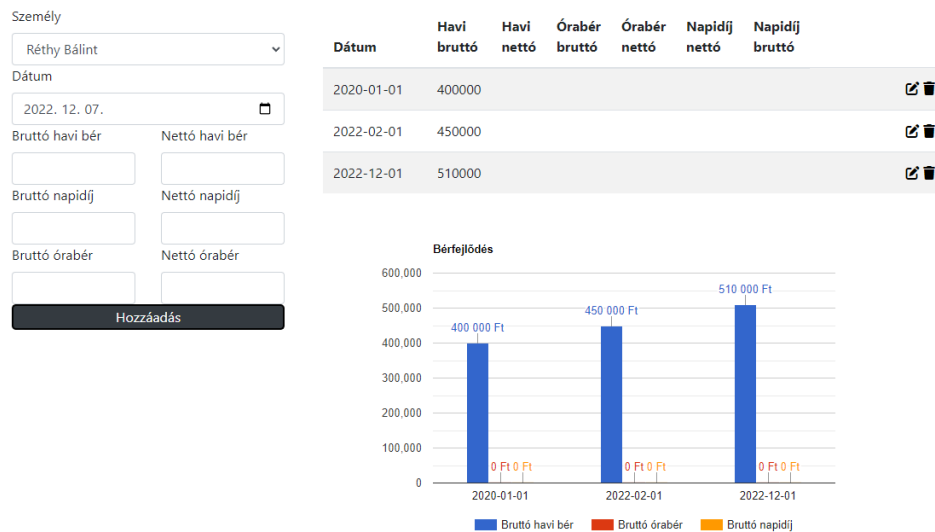
A pénzügyek menüpont alá tartoznak a bérek és a projekt költségek. Ezek alapján lehet később kimutatásokat készíteni a projektek megtérüléséről is.

Bérek

A bérek felület a személy kiválasztásával kezdődik. Ezután egy listában megjelennek a személy bérei az érvénybe kerülés dátumával együtt. Lehetőség van megadni bruttó és nettó havi bért, napidíjat, valamint órabért, viszont legalább az egyik mező töltése

kötelező. A hozzáadott bérek értékei alapján egy oszlopdiagram formájában rajzolódik ki a bérfelföldés. A bérek kezelésére a 34. ábrán láthatunk példát.

Bérek kezelése



34. ábra Bérek áttekintése

Költségek

A költségek felület a projektek költségeit segít áttekinteni. A projekt kiválasztása után listázásra kerülnek a költségek megnevezése, összegük, valamint keletkezésük dátuma. A 35. ábrán egy példa projekt költségeinek listázó oldalát láthatjuk.

Költségek listázása

Új költség
Projekt
Informatikai fejlesztés

Megnevezés	Összeg	Dátum
Notebook	250000	2020-02-15
Szervergép	500000	2020-01-20
Monitor	70000	2020-01-20

35. ábra Költségek áttekintése

5.4.3. Napi tevékenységek

Ide tartozik a napi munkaidő beosztás, a szabadságok kezelése, valamint a betegszabadságok kezelése.

Napi munkaidő beosztás

A napi munkaidő beosztás az alkalmazás leggyakrabban használt felülete. Itt lehet a dolgozókat a projektekhez rendelni és megadni, hogy hány órát dolgoztak az adott

projekten. Alapértelmezetten az aktuális napi dátummal nyílik meg a felület és automatikusan listázza az adott napra beosztott személyeket. A munkaszüneti napot egy jelölőnégyzet bepipálásával lehet jelölni. Ezt figyelembe fogja venni a bérköltségek és statisztikák számításakor. A projekt kiválasztása után lehetőség van megadni, hogy a kijelölt személyek melyik autóval utaztak, ha külső területen történik a munkavégzés. Amennyiben a cég saját telephelyén dolgoznak, vagy nem autóval közlekedtek, akkor az autó nélkül jelölőnégyzet bepipálásával ez rögzíthető. A személyek elkülönítve kerülnek listázásra, az első listában a cég belsős dolgozói jelennek meg, míg a második listában a külsős dolgozók. A személyekre kattintva automatikusan kijelöli őket, újbóli kattintásra törli a kijelölést. Ez a JavaScriptben megvalósított funkció növeli a felhasználói élményt, mivel egyszerűen megvalósítható a többszörös kijelölés és a csoportos hozzáadás. Végezetül a munkaidő és az utazási idő megadása következik. A hozzáadásra kattintva egyből frissül napi beosztás lista. A csoportos hozzáadás során előfordulhat, hogy a validációk miatt néhány személyt nem lehet projekthez rendelni, pl.: a személy szabadságon van. Ilyenkor azokat a hozzárendeléseket elvégzi a rendszer, amelyek helyesek, a helytelen hozzárendelésekről pedig visszajelzi a hibát a felhasználónak.

Hozzáadáskor legalább egy személy kiválasztása kötelező, valamint további validáció történik az üzleti logikai rétegben. Ilyen a dátum ellenőrzése, a projekt és a munkaóra kötelezősége, valamint a munkaóra és az utazási idő nem lehet negatív szám. Ezen kívül azok a személyek, akik szabadságon vagy betegszabadságon vannak nem rendelhetők projekthez.

A napi munkaidő beosztás kezelő felületére a 36. ábrán láthatunk példát.

Napi munkaidő beosztás

- Egyed Vince már hozzá lett adva 2022-12-06 0:00:00 dátummal szabadsághoz.

Dátum

☐ Munkaszüneti nap

Munka

Autó
☐ Autó nélkül

Belsős dolgozók

Réthy Bálint
Kozma Róbert
Egyed Vince
Oláh Donát

Külsős dolgozók

Kis Rudolf

Munkaidő

Utazási idő

Hozzáadás

Személy	Munka	Autó	Munka	Útidő
Réthy Bálint	Webalkalmazás fejlesztése		8	0
Kozma Róbert	Webalkalmazás fejlesztése		8	0
Oláh Donát	Webalkalmazás fejlesztése		8	0
Nagy Eufrozina	Webalkalmazás fejlesztése		8	0
Kis Rudolf	Tanácsadás	Ford Mondeo-GWP-090	8	0
Varga Barbara	Tanácsadás	Suzuki Swift-NUK-680	8	2
István Bence	Tanácsadás	Suzuki Swift-NUK-680	8	2
Kis Rudolf	Tanácsadás	Suzuki Swift-NUK-680	8	2

36. ábra Napi munkaidő beosztás

Szabadságok és betegszabadságok kezelése

A szabadságok és betegszabadságok kezelése felület hasonló a napi munkaidő beosztás felülethez. Két dátum mezővel kezdődik, melyek a kezdő és vég dátum, melyekkel több napig tartó szabadságot vagy betegszabadságot lehet hozzáadni. Alapértelmezetten a kezdő dátum az aktuális napi dátum. A felület betöltésekor listázza a kezdő dátum értékével hozzáadott személyeket, valamint a hozzájuk tartozó összes és hátralevő szabadságot vagy betegszabadságot. A többszörös kijelölés itt is engedélyezett és a személyek itt is belső és külső dolgozókra vannak bontva.

A szabadságok kezelését megvalósító felület a 37. ábrán, a betegszabadságokat kezelő felületet a 38. ábrán láthatjuk.

Szabadságok kezelése

Kezdő Dátum	Vég Dátum	Személy	Hátralevő szabadnapok	Összes szabadnap
2022-12-06	2022-12-06	Egyed Vince	21	22

Belső dolgozók

- Réthy Bálint
- Kozma Róbert
- Egyed Vince
- Oláh Donát

Külső dolgozók

- Kis Rudolf

Hozzáadás

37. ábra Szabadságok kezelése

Betegszabadságok kezelése

Kezdő Dátum	Vég Dátum	Személy	Hátralevő betegszabadságok	Összes betegszabadság
2022-01-20	2022-01-20	Réthy Bálint	15	15
		Kozma Róbert	15	15
		Egyed Vince	15	15
		Oláh Donát	15	15
		Kapolcs Zsanett	15	15
		Balla Hajnalka	15	15
		Nagy Eufrozina	15	15
		Varga Barbara	15	15
		István Bence	15	15
		Győri Ferenc	15	15

Belső dolgozók

- Réthy Bálint
- Kozma Róbert
- Egyed Vince
- Oláh Donát

Külső dolgozók

- Kis Rudolf

Hozzáadás

38. ábra Betegszabadságok kezelése

5.5. Jogosultságok áttekintése

A felhasználókhöz szerepkörök rendelhetők. Ezt a hozzárendelést csak a superadmin és a további adminisztrátorok tudják megtenni. Az alkalmazáshoz hozzáadott szerepköröket és egy felhasználóhoz való hozzárendelést a 39. ábra mutatja be.

Szerepkörök hozzárendelése

Email cím

Szerepkörök

Hozzárendelés

39. ábra Szerepkörök hozzárendelése

A felhasználók a megadott szerepkörök alapján érik el a funkciókat. A jogosultságok eloszlását a 40. ábra szemlélteti.

Kategória	Funkciók	Szerepkörök			
		Adminisztrátor	Részlegvezető	Pénzügy	Többi felhasználó
Törzsadatok	Törzsadatok megtekintése	X	X	X	X
	Törzsadatok hozzáadása	X	X	X	X
	Törzsadatok törlése	X			
	Személyek módosítása	X			
	Cégek módosítása	X			
	Projektek módosítása	X	X		
	Autók módosítása	X	X		
	Beosztások módosítása	X			
Pénzügyek	Címek módosítása				
	Bérek megtekintése	X		X	
	Bérek hozzáadása	X		X	
	Bérek módosítása	X			
	Bérek törlése	X			
	Költségek megtekintése	X	X	X	
	Költségek hozzáadása	X	X	X	
	Költségek módosítása	X	X	X	
Napi munka Szabadság betegszabadság	Költségek törlése	X	X	X	
	Napi tevékenységek megtekintése	X	X	X	
	Napi tevékenységek hozzáadása	X	X		
	Napi tevékenységek módosítása	X	X		
	Napi tevékenységek törlése	X	X		

40. ábra Jogosultság mátrix

5.6. Kimutatások készítése

A kimutatások célja, hogy áttekintést adjon a projektek bevételei és költségei, valamint a dolgozók munkabére tekintetében. A kimutatásokhoz szükséges adatok frissítése jelenleg kézzel indítható a menüből. A frissítésnek 3 módja lett implementálva.

Az első módszer a „Minden adat frissítése V1” gombra kattintva indítható. Ez egy táblába gyűjti össze a béreket és a költségeket és ebből rajzolt ki a felület egy Google Chart diagramot. A frissítéshez használt algoritmus először lekérdezi az összes költséget, az

összes napi munka ráfordítást, a személyek béreit, végezetül pedig az aktív projekteket. Ezután egy közös entitás típus lett bevezetve, a WageOrExpense (*bért vagy projekt költséget tartalmazó entitás*). Egy listába össze lett gyűjtve az összes projekt költség, valamint a napi munka ráfordításból adódó személyek bérköltségei. Itt fontos megemlíteni, hogy ha az órabér ki van töltve, akkor azzal számol, egyébként a napi díjjal, ha pedig az sincs megadva, akkor a havi fizetéssel. A bérköltségeknél mindig a napi munkavégzés dátumakor aktuális bér van figyelembe véve. Ha az adott nap munkaszüneti nap, akkor egy 1,5-szeres szorzóval számol. A lista tartalma végül adatbázisba lesz mentve és ebből rajzolódik ki a diagram.

Ez a frissítés és kirajzolás időigényes volt, ezért további szétbontás és optimalizálás volt szükséges, valamint a nagy mennyiségű adat miatt a havi bontású megjelenítés átláthatóbb eredményt hozott.

A hatékonyabb módszer a „Minden adat frissítése V2” menüpontra érhető el. Ez a módszer több segédtáblát használ az áttekinthetőbb számítások és a gyorsabb lekérdezés érdekében. Az algoritmus az első verziót felhasználva készült el. A béreket és költségeket tartalmazó közös lista tartalma itt is adatbázisba íródik, majd ebből készül egy napi összesítés projektenként, amely a DailyWageOrExpense nevet kapta. A havi bontású adatok előállításához végig kell iterálni a napi összegzésű adatokon és csoportosítani év, hónap és projekt alapján, valamint összegezni a költségeket. Az így előállt lista MonthlyWageOrExpense entitásokat tartalmaz, ez is adatbázisba íródik és ez lesz a forrása a főoldalon látható projekt áttekintő diagramnak.

A harmadik opció a „Gyors frissítés”, itt csak az utolsó 2 hónap adatai frissülnek. A legtöbb esetben nem szükséges minden adat frissítése, így ez a funkció lesz a leggyakrabban használva.

6. TESZTELÉS

A fejlesztés elengedhetetlen része a tesztelés, aminek köszönhetően felderíthetők a hibák és ellenőrizhető a helyes működés. A következőkben 3 féle tesztelés lesz bemutatva, melyek az egységtesztelés, a felületi tesztelés, valamint a stressztesztelés.

6.1. Egységtesztek

Az üzleti logika műveleteinek tesztelésére, a hibás működés kiszűrésére és az adathelyesség biztosítására 160 egységteszt lett írva. Ez entitásonként 12-18 tesztet jelent, amelyek a CRUD műveleteket és egyéb üzleti szabályoknak való megfelelést vizsgálják. Ezek az entitások a személyek, cégek, projektek, autók, beosztások, címek, bérek, költségek, napi munkaidő beosztások, szabadságok, valamint betegszabadságok kezelése.

Először a minden entitás esetén megvalósított tesztek fogom részletezni, majd az entitás típusonként egyedi vizsgálatokat.

6.1.1. Minden entitás esetén megvalósított vizsgálatok

Az olvasás műveletnél vizsgálva van, hogy az olvasás eredménye megfelelő típusú entitást adott-e vissza.

A létrehozás műveletnél ellenőrzésre kerül, hogy a bemeneti adatok alapján a létrejött új rekord a megfelelő értékeket tartalmazza-e. Ezen kívül vizsgálva van, hogy az azonosító, a létrehozás dátuma, továbbá a rögzítő felhasználó azonosítója nem egyezik meg a bemeneti adatokkal, mivel ez a 3 mező értékét az üzleti logika automatikusan tölti ki.

Módosítás esetén a bemeneti adatok és a módosított rekord mezői alapján történik vizsgálat. Ezen kívül a módosításért felelős technikai mezők is tesztelésre kerülnek, mégpedig a módosítás dátuma és a módosítást végző felhasználó azonosító nem egyezhet meg a bemeneten és a módosított rekordon. Ezt a 2 mezőt szintén az üzleti logika tölti ki automatikusan.

A törlés folyamatának tesztelésekor szintén vizsgálva vannak a technikai mezők, melyek a törlés dátuma és a törlést végző felhasználó azonosítója. Ezen kívül ellenőrzésre kerül a törölni kívánt rekord is, hogy létezik-e, valamint a törlést végző felhasználó azonosítója is, melynek megléte kötelező a törlés folyamatához.

6.1.2. Egyedi vizsgálatok entitás típusonként

Személy esetén vizsgálva vannak a kötelezően töltött mezők. Ezek a név, a születési hely, a születési dátum, valamint a foglalkoztató cég. Továbbá nem lehet negatív szám a szabadság és a betegszabadság.

A cégek hozzáadásakor vizsgált kötelező mezők a név és a cím.

Projektek esetén a név, a megbízó cég, a kezdő dátum, valamint a határidő kötelezősége van vizsgálva. Továbbá a bevétel és a tervezett költség nem lehet negatív szám.

Autó esetén szintén a kötelező mezők vannak vizsgálva, mindhárom felületi mezője kötelező. Ezek a márka, a típus, valamint a rendszám.

A beosztásnál a minden entitás típushoz használt teszteken kívül csak a név kötelezősége van vizsgálva.

Cím esetén ellenőrzésre kerülnek a kötelező mezők, melyek az ország, az irányítószám, a város és a cím. Ezen kívül az irányítószám esetén vizsgálva van, hogy ha az irányítószám egy szám, akkor nem lehet negatív.

Bér esetén kötelezően töltött mezők kerülnek tesztelésre. Ezek a személy azonosítója, a kezdő dátum, valamint legalább az egyik bér típus bruttó vagy nettó értéke.

Költségek tesztelésekor a vizsgált kötelező mezők a név, az összeg, valamint a keletkezés dátuma. Ezen kívül az összeg nem lehet negatív szám.

A napi munkaidő beosztás tesztek vizsgálják, hogy a kötelezők mezők töltve vannak-e, melyek a személy azonosító, a dátum, a projekt hozzárendelés, valamint a munkaóra. Ezen kívül a munkaóra és az utazási idő nem lehet negatív szám, valamint a hozzáadott személy nem lehet szabadságon vagy betegszabadságon.

A szabadság és betegszabadság entitásoknál a napi munkához hasonlóan vizsgálják a tesztek, hogy a személy azonosító és a dátum ki vannak-e töltve. Továbbá ellenőrzik a tesztek, hogy az adott személy már hozzá van-e adva szabadsághoz, betegszabadsághoz, vagy hozzá van-e már rendelve projekthez a napi munkaidő beosztásnál.

6.2. Felületi tesztek

Szisztematikusan végigteszteltem az alkalmazás felületeit először helyes, majd helytelen adatokkal. Ezután a kimutatásokat vizsgáltam meg.

A bérek kezelése felületén meg lehetett adni negatív összeget, ez a vizsgálat hiányzott. Újratesztelés után a helytelen adat hozzáadása sikertelen és a hibát kijelzi a felhasználónak. Ezen kívül a bérek listázó felületén a bruttó napidíj és nettó napidíj címkéje fel volt cserélve.

A személy, a projekt, valamint a bér létrehozásakor hiányzott a hiba jelzése a felületen. A személyek, a projektek, az autók, a bérek, a költségek módosítása esetén szintén nem volt hibajelzés.

Hogyha a napi tevékenységek alá tartozó felületeknél a helytelen adat miatt visszajelzett hiba szövege tartalmazott dátumot, akkor a dátum nem rövid formátumban volt kiírva.

A személy, a projekt, a költség, valamint a napi munkaidő beosztás szerkesztő felületén a dátum mezők visszatöltése sikertelen volt.

A fent említett hibák javítása megtörtént, újratestelésük után megfelelő volt a működés.

A felületen megjelenített vonal és oszlopdiagramok értékeinek megjelenítése nem egységes. Előfordul ezresenként tagolás és vesszővel tagolás is, valamint a főoldalon szereplő projekt áttekintés a nagy értékek ezreseit K-val jelöli.

A főoldalon szereplő projekt áttekintése diagram mellett megjelenített lista tartalmazza az adott napra a projektre beosztott dolgozókat. Ez a lista megjelenítése bizonyos esetekben belecsúszott a diagram jelmagyarázatába, ezért áthelyezésre került.

Különféle szerepkörökkel rendelkező felhasználókkal is történt tesztelés. Néhány Controller Actionról hiányzott a szerepkör meglétét ellenőrző attribútum. Ezek pótolva lettek és az újból elvégzett tesztelés során megfelelő volt a működés.

6.3. Stressztesztek

Az alkalmazáshoz 2 évnyi teszt adat lett generálva Excelben. Ezek további átalakításával előálltak a DbContext-ben rögzített adatok, melyek az alkalmazás első indulásakor íródnak az adatbázisba.

Az így létrejött adatok a következők:

- 4 felhasználó, melyek egy superuser, egy adminisztrátor, egy pénzügyi felhasználó, valamint egy részlegvezető
- 4 szerepkör, melyek a felhasználókhöz lettek rendelve
- 8 autó
- 217 ország
- 6 cég és a hozzájuk tartozó 6 helyszín
- 7 beosztás
- 5 projekt és 7 projekt költség
- 11 személy és a hozzájuk tartozó bérek
- 450 szabadnap és 150 betegszabadság hozzárendelés
- végül pedig 7230 napi munkaidő beosztás

Ezekkel az adatokkal tesztelhetőek a listázó és szerkesztő felületek, valamint kimutatások is készíthetőek.

A teszteléshez használt konfiguráció Intel i5 6500 processzor, 16 GB RAM 2133 MHz, Samsung SSD 860 EVO, valamint 64-bites Windows 10 operációs rendszer.

A minden adat frissítése funkció 3 perc 42 másodperc alatt futott le. A frissítés eredményeként 7097 WageOrExpense (*bérköltség vagy projekt költség*), 741 DailyWageOrExpense (*bérköltség és projekt költség napi szinten csoportosítva és*

összegezve), valamint 38 MonthlyWageOrExpense (*bérköltség és projekt költség havi szinten csoportosítva és összegezve*) rekord jött létre. Az így előállított adatokból a főoldalon megvalósított kimutatások azonnal hoznak eredményt.

A gyors frissítés funkció csak az utolsó 2 hónap adatait frissíti, ez 1 másodpercig tartott.

7. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az alkalmazást több területen lehet továbbfejleszteni és finomhangolni. Az egyik legfontosabb még hiányzó funkció a naplózás. Minden rekord hozzáadásakor rögzítésre kerül a létrehozás dátuma és a hozzáadást végző felhasználó azonosítója, de emellett egy külön táblában lehetne naplózni a felhasználói interakciókat, hogy melyik felületen milyen adatváltozást végeztek. A naplózás másik fontos területe a programhibák rögzítése. Ezeket a hibákat ajánlott automatikusan elküldeni a fejlesztő csapatnak, hogy ők ezeket kivizsgálhassák és javíthassák.

A felületek és funkciók terén is lehetne növelni a felhasználói élményt. A címek csak a cégek hozzáadásakor kerülnek felhasználásra, viszont ezeket külön felületen létre kell hozni, ezután érhetőek el az új cég hozzáadása felületen. A cégek létrehozó felületén szerepelhetnének a cím mezői és kitöltés után a cím entitások is rögzítésre kerülnének. A másik opció a cég entitás kibővítése a cím mezőivel.

A napi munkaidő beosztás kezelésekor lehetőség van jelölni a munkaszüneti napot. Ha egy adott nap hétvégére esik, akkor a munkaszüneti nap jelölőnégyzet lehetne alapértelmezetten bejelölve, de továbbra is módosítható, ha például hétfői munkanap van.

Bizonyos funkciókat lehetne korlátozni, hogy csak intranetről legyenek elérhetőek vagy VPN-ről (*Virtual Private Network – virtuális magánhálózat*). Az internetes elérés korlátozódhatna az adatok olvasására, míg az adatok módosítása, törlése csak intranetről. További lehetőség, hogy az adminisztrátoroknak internetről is legyen joguk minden művelethez.

A kimutatásokhoz szükséges adatok frissítése jelenleg manuálisan történik egy menüpontra kattintva. Ezt a végpontot meg lehetne hívni parancssorból curl (*több protokollt támogató parancssori eszköz*) hívással, ezt a parancsfájlt pedig a Windows feladat ütemezővel automatizálni. Így garantálható a rendszeres frissítés és lehetőség van például éjszakai frissítésre.

A CQRS továbbfejlesztése is növelné a hatékonyságot. A parancskezelők bevezetésével az adatváltozások naplózva lennének. Ezek a műveletek később egy másik adatbázison reprodukálhatóak, így a lekérdezési adatbázis adatai is konzisztensek lesznek, valamint nem rontja az operatív adatbázis teljesítményét.

Az adatok törlése logikai törlésként van megvalósítva. Ez azt a célt szolgálja, hogy ha egy adat véletlen törlésre kerül, akkor lehetőség van azt visszaállítani. A visszaállításhoz meg lehetne határozni egy időkorlátot, például 30 napot, ezután az adat véglegesen törlődik, esetleg egy másik táblában vagy adatbázisban rögzítésre kerül.

8. ÖSSZEFOGLALÁS

Ebben a dolgozatban bemutatásra kerül az ERP használatának fontossága. Először általános tudnivalókat ismertet, majd az elterjedését hátráltató tényezőket, valamint a használatának előnyeit mutatja be.

Ezután egy ERP bevezetéseinek lépései következnek, részletezi a fontos teendőket, melyek cégenként eltérőek lehetnek. A továbbiakban a piacon fellelhető fizetős és ingyenes vállalatirányítási információs rendszerek bemutatása valósul meg. A következő rész a terheléselosztást taglalja, mely hozzájárul az egyenletes terhelés biztosításához, növeli a teljesítményt és a rendelkezésre állást. Az ismertebb terheléselosztó algoritmusok is bemutatásra kerülnek. A hardveres és szoftveres terheléselosztás jellemzői, valamint az összehasonlításuk is ismertetésre kerülnek. Itt fontos megemlíteni a parancs-lekérdezés felelősség elkülönítésének módszerét, amelyre a megvalósítás során támaszkodik az alkalmazás.

Az irodalomkutatás után a rendszerterv, amely előre vetíti a megvalósítandó funkciókat. Ezután a felhasznált módszerek és technológiák, valamint azok indoklása olvashatóak.

Ezt követi a megvalósítás, amely részletesen specifikálja a funkciókat és képernyőképekkel illusztrálja az elkészült alkalmazás felületeit. Kitér a különböző területek specialításaira, amelyek növelik a felhasználói élményt, valamint biztosítják, hogy ne kerülhessen helytelen adat a rendszerbe.

Végezetül a rendszer tesztelésének dokumentálása olvasható, mely során 3 féle tesztelés valósult meg. Először az egységtesztek kerülnek bemutatásra, melyek az üzleti logika műveleteit és egyéb üzleti szabályokat vizsgálnak. Ezután a felületi tesztek következnek, melyekhez az alkalmazás használata során gyűlt tapasztalat. Utolsó pontként pedig a stressztesztek mutatják be, hogy nagy mennyiségű adat esetén mennyi időt vesz igénybe a kimutatásokhoz szükséges adatok frissítése, valamint milyen válaszidővel tekinthetők meg ezek a kimutatások.

9. ABSTRACT

In this thesis, the importance of using ERP is presented. It will first provide a general information, and then discuss the barriers to its uptake and the benefits of its use.

Next, it goes on to describe the steps of implementing an ERP, detailing the important tasks, which may vary from company to company. It then presents a selection of paid and free ERP systems available on the market. The next section discusses load balancing, which helps to ensure a balanced workload, increase performance and availability. The more well-known load balancing algorithms are also presented. The characteristics of hardware and software load balancing and their comparison are also described. It is important to mention here the Command-Query Responsibility Segregation method that the application relies on for its implementation.

After the literature research, the system design, which anticipates the features to be implemented. This is followed by the methods and technologies used and their justification.

This is followed by the implementation, which specifies the functions in detail and illustrates the interfaces of the completed application with screenshots. Special features in the different areas are highlighted to increase the user experience and to ensure that no incorrect data is inserted into the system.

Finally, it documents the testing of the system, which involved 3 types of testing. First, unit tests are presented, which test the business logic operations and other business rules. Next, the UI tests, which are based on the experience gained during the use of the application. Lastly, stress tests show how long it takes to update the data required for reports when large amounts of data are involved, and the response time to view these reports.

10. IRODALOMJEGYZÉK

- [1] Wikipédia – Vállalatirányítási információs rendszerek
(https://hu.wikipedia.org/wiki/V%C3%A1llalatir%C3%A1ny%C3%ADt%C3%A1si_inform%C3%A1ci%C3%B3s_rendszerek), utoljára megtekintve: 2022.10.03.
- [2] Sneller, L.: A Guide to ERP: Benefits, Implementation and Trends. *bookbon.com*, 2014
- [3] Az ERP bevezetésének 10 kritikus pontja
(http://www.revolution.hu/demo/az_erp_bevezetesek_10_kritikus_pontja.pdf), utoljára megtekintve: 2022.10.30.
- [4] Az ERP rendszerek elavultak?
(<https://gobertpartners.com/are-erp-systems-obsolete>), utoljára megtekintve: 2022.10.30.
- [5] Mi az ERP? (Enterprise Resource Planning)
(<https://www.netsuite.com/portal/resource/articles/erp/what-is-erp.shtml>), utoljára megtekintve: 2022.10.30.
- [6] Az ERP rendszer bevezetés 6 fázisa
(<https://clouderp.hu/vallalatiranyitasi-rendszer/erp-rendszer-bevezetes>), utoljára megtekintve: 2022.10.06.
- [7] Legnépszerűbb ERP rendszerek
(<https://softwareconnect.com/erp/erp-market/>), utoljára megtekintve: 2022.10.08.
- [8] Microsoft Dynamics 365 üzleti alkalmazások
(<https://adynamics.com.my/applications/microsoft-dynamics-365>), utoljára megtekintve: 2022.10.12.
- [9] Mi az SAP és miért fontos?
(<https://www.corealm.com/what-is-sap-why-is-it-important>), utoljára megtekintve: 2022.10.12.
- [10] Deltek Vantagepoint részletes áttekintés
(<https://www.erp-information.com/deltekvantagepoint>), utoljára megtekintve: 2022.10.13.
- [11] Sage 300 Cloud és Sage Business összehasonlítása
(<https://www.trustradius.com/compare-products/sage-300cloud-vs-sage-business-cloud-accounting>), utoljára megtekintve: 2022.10.14.
- [12] Sage 300 és Sage 500 összehasonlítása
(<https://www3.technologyevaluation.com/selection-tools/comparison->

[reports/22960/sage-500-erp-vs-sage-300-erp-for-distribution-industries-compare](#)), utoljára megtekintve: 2022.10.14.

[13] Oracle vállalatirányítási információs rendszer
(<https://www.oracle.com/erp/>), utoljára megtekintve: 2022.10.31.

[14] Legnépszerűbb ingyenes ERP-k és nyílt forráskódú rendszerek, valamint rejtett költségeik
(<https://www.erpfocus.com/top-free-erp-systems.html>), utoljára megtekintve: 2022.10.31.

[15] 2022-ben elérhető legjobb ERP szoftverek
(<https://www.thesmbguide.com/free-erp-software>), utoljára megtekintve: 2022.10.20.

[16] Minden, amit érdemes tudni a Bitrix24-ről
(<https://www.bitrix24.com/articles/everything-you-need-to-know-about-bitrix24.php>), utoljára megtekintve: 2022.10.22.

[17] Biztonságos kikötő jogszabály bemutatása
(<https://peterswire.net/archive/pssafe~1.htm>), utoljára megtekintve: 2022.10.22.

[18] Bitrix24 szoftver árazásának bemutatása
(<https://www.bitrix24.com/prices/>), utoljára megtekintve: 2022.10.22.

[19] iDempiere ERP bemutatása, előnyök, hátrányok, valamint az ADempiere, iDempiere és Odoo kapcsolata
(<https://www.erp-information.com/idempiere-erp-details>), utoljára megtekintve: 2022.11.01.

[20] OSGi áttekintése
(<https://www.ibm.com/docs/en/wasdtfe?topic=applications-osgi-overview>), utoljára megtekintve: 2022.11.01.

[21] ADempiere áttekintése
(<https://web.archive.org/web/20100420032819/http://www.adempiere.org/home/aboutus>), utoljára megtekintve: 2022.11.01.

[22] iDempiere áttekintése
(<https://www.idempiere.org/about/>), utoljára megtekintve: 2022.10.24.

[23] Odoo visszajelzések
(<https://www.thesmbguide.com/odoo-reviews>), utoljára megtekintve: 2022.11.03.

[24] Az Odoo története
(https://www.odoo.com/blog/odoo-news-5/the-odoo-story-56?blog=blog.blog%285%2C%29&blog_post=blog.post%2856%2C%29), utoljára megtekintve: 2022.11.03.

- [25] A Big 4 könyvelő cégek
(<https://big4accountingfirms.org/>), utoljára megtekintve: 2022.11.03.
- [26] Odoo árazásának áttekintése
(<https://www.odoo.com/pricing-plan>), utoljára megtekintve: 2022.11.03.
- [27] Odoo Studio
(<https://crozdesk.com/software/odoo-studio>), utoljára megtekintve: 2022.11.03.
- [28] Terheléselosztás jellemzői
(<https://www.cloudflare.com/learning/performance/what-is-load-balancing>), utoljára megtekintve: 2022.11.05.
- [29] Terheléselosztás jellemzői
(<https://avinetworks.com/what-is-load-balancing>), utoljára megtekintve: 2022.11.05.
- [30] Round-Robin DNS bemutatása
(<https://www.cloudflare.com/learning/dns/glossary/round-robin-dns>), utoljára megtekintve: 2022.11.06.
- [31] Terheléselosztó algoritmusok és technikák
(<https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques>), utoljára megtekintve: 2022.11.13.
- [32] IP Hash terheléselosztási módszer
(<https://kemptechnologies.com/glossary/source-ip-hash-load-balancing>), utoljára megtekintve: 2022.11.13.
- [33] IP Hash terheléselosztási módszer
(<https://blogs.infoblox.com/community/source-ip-hash-load-balancing-for-application-persistence>), utoljára megtekintve: 2022.11.13.
- [34] Szoftveres és hardveres terheléselosztás összehasonlítása
(<https://www.cdw.com/content/cdw/en/articles/networking/how-to-choose-between-software-and-hardware-based-load-balancers.html>), utoljára megtekintve: 2022.11.19.
- [35] Szoftveres és hardveres terheléselosztás összehasonlítása egy ERP szempontjából
(<https://www.techtarget.com/searchitoperations/answer/Hardware-vs-software-load-balancer-Which-is-better-for-an-enterprise>), utoljára megtekintve: 2022.11.19.
- [36] Rack és Stack szerkezet ismertetése
(<https://www.racksolutions.com/news/blog/rack-and-stack>), utoljára megtekintve: 2022.11.19.
- [37] Rack és Stack és a kapcsolódó architektúra bemutatása
(<https://imstechsec.com/rack-stack>), utoljára megtekintve: 2022.11.19.

- [38] Parancs-lekérdezés felelősség elkülönítése
(<https://www.techtarget.com/searchapparchitecture/definition/CQRS-command-query-responsibility-segregation>), utoljára megtekintve: 2022.11.27.
- [39] Parancs-lekérdezés felelősség elkülönítése
(<https://martinfowler.com/bliki/CQRS.html>), utoljára megtekintve: 2022.11.28.
- [40] Parancs tervezési minta
(<https://refactoring.guru/design-patterns/command>), utoljára megtekintve: 2022.11.29.
- [41] Saga elosztott tranzakciók mintája
(<https://learn.microsoft.com/hu-hu/azure/architecture/reference-architectures/saga/saga>), utoljára megtekintve: 2022.11.29.
- [42] Jelentési adatbázis
(<https://martinfowler.com/bliki/ReportingDatabase.html>), utoljára megtekintve: 2022.11.29.
- [43] Vastag- és vékonykliens összehasonlítása
(<https://itjunction.org/2019/10/05/what-is-the-difference-between-thick-client-and-thin-client>), utoljára megtekintve: 2022.11.30.
- [44] Rétegelt webalkalmazás jellemzői
(<https://www.cleveroad.com/blog/web-application-architecture>), utoljára megtekintve: 2022.12.02.
- [45] SQL Express áttekintése és előnyei ismertetése
(<https://neovera.com/sql-server-express-use>), utoljára megtekintve: 2022.12.05.
- [46] SQL szerver variációk
(<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>), utoljára megtekintve: 2022.12.05.