



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR



SZAKDOLGOZAT

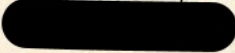
OE-NIK
2024

Hallgató neve:
Hallgató törzskönyvi száma:

Szilágyi Ferenc
T/008577/FI12904/N

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertervezés és -fejlesztés Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Szilágyi Ferenc**
Törzskönyvi száma: T/008577/FI12904/N
Neptun kódja: 

A dolgozat címe:

Fiskaly DSFinV-K API integrálása a LS Central-ba
Integration of Fiskaly DSFinV-K API into LS Central

Intézményi konzulens: Sipos Miklós László
Külső konzulens:

Beadási határidő: 2024. május 15.

A záróvizsga tárgyai: Számítógép architektúrák
Big Data és üzleti intelligencia
specializáció

A feladat

A feladat létrehozni egy lokális Microsoft Dynamics 365 Business Central DE környezetet a legfrissebb verzióval PowerShell segítségével. Importálja demo W1 adatokat konfigurációs csomagok segítségével. Készítsen egy adatbázis backup-ot. Publikálja a szükséges függőségeket az LS Central német lokalizációjához. Regisztráljon egy fiskaly fiókot és készítse el az API key-t és secret-et. Postman segítségével tetsztelje a kapcsolatot és elemezze a végpontokat. Elemezze a már megvalósított Sign DE V2 API-t és keressen újrafelhasználható kódot. VS Code segítségével AL nyelven integrálja a DSFinV-K szükséges elemeit. Alakítsa ki a master-slave hierarchia rendszert a pénztárak között. Kösse össze a tranzakciókat a pénztárakkal. Készítse elő a pénztárzárást, használjon staging rekordokat. Készítsen feladatvárólistát, ami az egyes rekordok státuszát frissíti. Integrálja az új rekordokat a meglévő felhasználói felületre, illetve készítsen oldalakat is. Oldja meg, hogy a funkcionalitás kapcsolható legyen üzlethelyiségenként. Fejlesztés közben győződjön meg róla, hogy az adatok frissülnek a fiksaly Dashboard felületén. Kezdeményezze a pénztárzárási exportot Business Central-on keresztül. Fiskaly Dashboardon keresztül kérje le az elkészült csv audit fájlt és ellenőrizze, hogy helyes adatokat tartalmaz-e.

A dolgozatnak tartalmaznia kell:

- a megoldandó feladat pontos leírását és részletes elemzését,
- magas szintű szerkezeti mintát az alkalmazásról,
- az egyes funkciók megvalósításához felhasználható lehetséges technológiákat,
- az elkészített rendszer tervét, blokkdiagramját,
- a megvalósítás menetét, felmerülő problémákat és az azokra adott megoldást,
- a tesztelési tervet, a tesztelés módszertanát és a tesztelés eredményeit,
- az elkészült szoftver felhasználási lehetőségeit,
- az eredmények bemutatását és értékelését a teljes munkára vonatkozóan.



Vámosy Z

Dr. habil. Vámosy Zoltán
intézetigazgató

A szakdolgozat elévülésének határideje: **2026. május 15.**
(ÓE HKR 54.§ (10) bekezdés szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

Sipos
.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

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

Budapest, 2024.12.06

hallgató aláírása

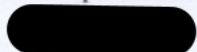


KONZULTÁCIÓS NAPLÓ

Hallgató neve:

SZILÁGYI FERENC

Neptun Kód:



Tagozat:

LEVELEZŐ

Telefon:

Levelezési cím (pl.: lakcím):



Szakdolgozat címe magyarul:

FISKALY DSFINV-K API INTEGRÁLÁSA A LS CENTRAL-BA

Szakdolgozat címe angolul:

INTEGRATION OF FISKALY DSFINV-K API INTO LS CENTRAL

Intézményi konzulens:

SIPOS MIKLÓS LÁSZLÓ

Alk.	Dátum	Tartalom	Aláírás
1.	2024.02.19.	Félév teendőinek, menetrendjének és mérföldköveinek ismertetése.	
2.	2024.03.11.	Hasonló rendszerek elemzése.	
3.	2024.04.15.	Irodalomkutatás, annak összegzése valamint konzekvenciák meghatározása.	
4.	2024.05.06.	Tervezés, követelmények specifikációja. Véglegesítés. Leadással kapcsolatos teendők egyeztetése.	

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

A hallgató a Szakdolgozat I tantárgy követelményét teljesítette, beszámolóra bocsátható.

A konzulens által javasolt érdemjegy: 5

Intézményi konzulens

Budapest, 2024. 05. 17.

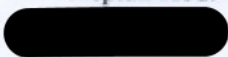


KONZULTÁCIÓS NAPLÓ

Hallgató neve:

SZILÁGYI FERENC

Neptun Kód:



Tagozat:

LEVELEZŐ

Telefon:

Levelezési cím (pl.: lakcím):



Szakdolgozat címe magyarul:

FISKALY DSFINV-K API INTEGRÁLÁSA A LS CENTRAL-BA

Szakdolgozat címe angolul:

INTEGRATION OF FISKALY DSFINV-K API INTO LS CENTRAL

Intézményi konzulens:

SIPOS MIKLÓS LÁSZLÓ

Alk.	Dátum	Tartalom	Aláírás
1.	2024.09.29.	Fejlesztés tervezett menetének revíziója, szakmai egyeztetés, fejlesztési teendők megkezdése	
2.	2024.10.20.	Fejlesztés állapotának és haladásának áttekintése. Technikai problémák és módosítások egyeztetése.	
3.	2024.11.17.	Felmerülő problémák egyeztetése és megoldása. Tesztelési fázis megindítása.	
4.	2024.12.01.	Dokumentáció véglegesítése. Leadással kapcsolatos teendők egyeztetése.	

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

A hallgató a Szakdolgozat II tantárgy követelményét teljesítette, védeésre bocsátható.

A konzulens által javasolt érdemjegy: 5

Intézményi konzulens

Budapest, 2024. 12. 06.

ABSZTRAKT

A szakdolgozatom célja a fiskaly DSFinV-K szolgáltatás integrációjának vizsgálata és megvalósítása Local Functionality for LS Central DE bővítményben, a német pénzügyi tranzakciós szabályozási követelményeknek megfelelően. Ennek érdekében áttekintettem és elemeztem a fejlesztési eszközöket és módszertanokat, majd ezek alapján döntést hoztam, hogy melyik felel meg legjobban a projekt igényeinek és célkitűzéseinek.

A dolgozatban részletesen ismertetem a fiskaly DSFinV-K szolgáltatásának működési szempontjait és a sikeres megvalósításához szükséges összes kritikus integrációs pontot. Meghatároztam a szükséges adatbázis-táblákat és felhasználói felület módosításait, amelyek elengedhetetlenek az adatok konzisztens kezeléséhez és a megfelelő felhasználói élmény biztosításához. A fejlesztés első lépéseként létrehoztam egy fejlesztői környezetet, amely biztosítja, hogy minden szükséges függőség rendelkezésre álljon a későbbi fejlesztésekhez.

A fejlesztési környezet előkészítését követően a tábla és felhasználói felület módosításait implementáltam. Ezek megvalósítása után készült el az a logikai réteg, amely az adatokat fogadja, kezeli és továbbítja a rendszerben. Az üzleti logikába való integrálás mellett egy feladatvárólistához adható programmodul is kialakításra került, amely lehetővé teszi bizonyos rekordok háttérben történő automatikus frissítését.

A projekt során unit tesztekkel végeztem, biztosítva, hogy az elkészült rendszer minden követelménynek megfeleljen, és stabil működést nyújtson a produkciós környezetben is.

ABSTRACT

The aim of my thesis is to investigate and implement the integration of the fiskaly DSFinV-K service in the Local Functionality for LS Central DE extension, in accordance with the German financial transaction regulatory requirements. To this end, I reviewed and analysed development tools and methodologies, and based on these, I decided which one best fits the needs and objectives of the project.

In this thesis, I will describe in detail the operational aspects of the fiskaly DSFinV-K service and all critical integration points required for its successful implementation. I have identified the necessary database tables and user interface modifications that are essential to ensure consistent data management and a proper user experience. As a first step in the development, I created a development environment to ensure that all necessary dependencies are available for future development.

After preparing the development environment, I implemented the changes to the table and user interface. After these were implemented, the logical layer that receives, manages and transmits the data in the system was completed. In addition to the integration into the business logic, a task queue add-on module was developed to allow automatic background updating of certain records. Throughout the project, I conduct unit and integration tests to ensure that the completed system meets all requirements and provides stable operation in the actual production environment.

During the project, I conducted unit tests to ensure that the finished system met all the requirements and provided stable operation in the production environment.

TARTALOMJEGYZÉK

1. Bevezetés	8
2. Irodalomkutatás.....	9
2.1. Hasonló rendszerek.....	9
2.1.1. POS rendszerek.....	9
2.1.2. Square	9
2.1.3. Shopify.....	9
2.1.4. Lightspeed.....	10
2.1.5. LS Central	10
2.1.6. Összefoglaló.....	10
2.2. Microsoft Dynamics 365 Business Central.....	11
2.3. Local Functionality for LS Central DE.....	11
2.4. Fiskaly.....	11
2.4.1. KassenSichV	11
2.4.2. DSFinV-K	11
2.4.3. RKSV.....	12
2.4.4. TicketBAI.....	12
2.4.5. Vari*factu	12
2.4.6. Fiskalcheck	12
2.4.7. Digital Receipt	12
2.5. Fiskaly API.....	12
2.5.1. Fiskaly SIGN DE V2	13
2.5.2. Fiskaly DSFinV-K.....	13
2.6. DSFinV-K végpontok	13
2.6.1. Authentication	13
2.6.2. VAT Definition.....	13
2.6.3. Purchaser Agency.....	13
2.6.4. Cash Register	14
2.6.5. Cash Point Closing.....	14
2.6.6. Exports	14
2.7. Verziókezelés	14
2.7.1. Git	14

2.7.2. Bitbucket.....	15
2.7.3. Azure DevOps.....	15
2.7.4. GitHub.....	15
2.7.5. GitLab	15
2.7.6. Összefoglaló.....	16
2.8. Parancssor	16
2.9. PowerShell	16
2.10. Powershell ISE.....	17
2.11. Business Central Administration Shell	17
2.12. Összegzés.....	17
2.13. Visual Studio Code	17
2.13.1. AL Language Extension for Microsoft Dynamics 365 Business Central	18
2.13.2. Egyéb bővítmények a produktivitás növelésére	18
2.14. Fejlesztői környezet	19
2.14.1. SaaS Sandbox.....	19
2.14.2. Docker konténer.....	20
2.14.3. Helyi telepítés	21
2.14.4. Összefoglaló.....	22
2.15. Tesztelés	23
2.16. API fejlesztői platform.....	24
2.16.1. Postman.....	24
2.16.2. Soap UI	24
2.16.3. Swagger UI	24
2.16.4. Összefoglaló.....	24
2.17. . Fejlesztési módszertanok	25
2.17.1. . Vízés modell	25
2.17.2. Agilis módszertanok.....	26
2.17.3. . Scrum	26
2.17.4. . Kanban	26
2.17.5. . Összefoglaló.....	27
3. Követelmény specifikáció.....	28
3.1. Előkövetelmény	28

3.2. Céges követelmény	28
3.3. Funkcionális követelmény	28
4. Tervezés	29
4.1. Rendszerterv	29
4.2. Funkciólista.....	31
4.2.1. Autentikáció	31
4.2.2. VAT Definition rekord létrehozása/módosítása	32
4.2.3. Cash Register létrehozása/módosítása	32
4.2.4. Cash Point Closing létrehozása.....	33
4.2.5. Cash Point Closing lekérdezése.....	35
4.2.6. Cash Point Closing törlése.....	35
4.2.7. Fiskaly Sales Transaction kiegészítése	35
4.3. Fejlesztés előkészítése	36
4.4. Tábla módosítások	36
4.4.1. Fiskaly Setup.....	36
4.4.2. Store	36
4.4.3. POS VAT Code.....	36
4.4.4. Fiskaly Sales Transaction.....	37
4.4.5. Cash Register	37
4.4.6. Cash Point Closing.....	37
4.5. Felhasználói felület	37
4.5.1. Fiskaly Setup.....	38
4.5.2. POS VAT Code.....	38
4.5.3. Store	38
4.5.4. POS Terminal Card	38
4.5.5. Új oldalak.....	38
4.6. Interfacek	38
4.7. Programmodulok.....	38
4.8. Fejlesztés tervezett menete	39
4.8.1. Analízis folyamat	39
4.8.2. Előkészítés	39
4.8.3. Meglévő táblák kiegészítése	39

4.8.4. Cash Registers (pénztárak)	39
4.8.5. Cash Point Closing (pénztárzárás)	40
4.8.6. DSFinV-K export	40
4.8.7. Feladatvárólista	40
4.8.8. Tesztelés	40
5. Fejlesztés	41
5.1. Analízis	41
5.2. Előkészítés	43
5.3. Fejlesztői környezet	43
5.4. Eljárások	44
5.5. Logic Factory	44
5.6. Autentikáció	45
5.7. Meglévő objektumok kiegészítése	46
5.7.1. Fiskaly Setup.....	46
5.7.2. POS VAT Code.....	46
5.7.3. Store	47
5.7.4. POS Terminal Card	47
5.8. Új objektumok	47
5.8.1. Cash Register	47
5.8.2. Cash Point Closing.....	48
5.9. Service Locator	49
5.10. Feladatvárólista	50
5.11. DSFinV-K export	50
6. Tesztelés	51
6.1. Manuális.....	51
6.2. Tesztautomatizálás	56
6.2.1. Autentikáció	57
6.2.2. Statement Posting.....	58
7. Értékelés.....	60
8. Továbbfejlesztési lehetőségek.....	61
9. Összegzés.....	62
10. Summary	63

1. BEVEZETÉS

Szilágyi Ferenc vagyok, hat éves AL fejlesztői tapasztalattal rendelkezem. Jelenleg az LS Retail ehf.-nél dolgozom, ahol a lokalizációs csapatot erősítem. Cégünk egyedi Point of Sale (POS) rendszert kínál, amely kiegészíti a Microsoft Dynamics 365 Business Central ERP funkcionalitását, és jelenleg több mint 110.000 üzletben van jelen a világ 156 országában [1].

A projekt fő célja a német lokalizáció továbbfejlesztése a DSFinV-K API integrációjával, amely biztosítja, hogy az LS Central alkalmazás megfeleljen a szükséges jogi követelményeknek. Korábban már sikeresen integráltuk a SIGN DE V2 API-t, amely lehetővé tette minden tranzakció fiskaly rendszerbe történő beküldését. Ennek érdekében egyedi Technical Security System (TSS) azonosítót rendeltünk minden üzlethez, valamint klienst hoztunk létre az üzletekhez tartozó terminálokhoz [2].

A fejlesztés során minden tranzakciót pénztárgéphez (Cash Register) társítok, és a napi zárás alkalmával lehetővé válik a pénztárzárás is, amely bármikor exportálható a fiskaly rendszerből. A Cash Register-eket terminálokhoz rendelem, és egy master-slave hierarchiát alakítok ki közöttük. A master Cash Register minden tranzakciót magába foglal, beleértve az alárendelt (slave) Cash Register által végrehajtott tranzakciókat is. Napi zárások (Close out Summary) alkalmával pénztárzárás (Cash Point Closing) is létrejön, amely képes automatikus vagy manuális működésre, hasonlóan a SIGN DE V2-hez.

A szakdolgozat során részletesen vizsgáltam a fiskaly rendszer működését, majd a tervezési szakaszt követően megvalósítottam az új funkcionalitást melyet manuálisan és automatikusan is teszteltem.

Feltételeztem, hogy az olvasó rendelkezik alapvető ismeretekkel az AL nyelvről és a Business Central rendszerről, ezért nem tértem ki részletesen a nyelvi elemek ismertetésére.

2. IRODALOMKUTATÁS

2.1. Hasonló rendszerek

Saját meglévő alkalmazásunk funkcionalitását új funkcionalitással egészítettem ki, így teljes mértékben egyedi megoldást készítettem. Az alábbiakban részletes összehasonlítást végeztem a fejlesztéshez szükséges eszközökről.

2.1.1. POS rendszerek

A kereskedelemben a POS-rendszerek (Point of Sale) a hagyományos pénztárgépekből modern elektronikus terminálokká fejlődtek, megkönnyítve a különböző fizetési módokat. Ezek a fizikai és online üzletekben egyaránt megtalálható rendszerek ma már fejlett szoftverekkel rendelkeznek, lehetővé téve a kiskereskedők számára a készletek nyomon követését, a vásárlási trendek elemzését, az árképzés pontosságának biztosítását és fontos marketingadatok gyűjtését. POS-tranzakciók mind a fizikai, mind az online üzletekben előfordulnak, a felhőalapú rendszerek pedig egyre népszerűbbek, különösen a nagy online kereskedők körében. Ezek a rendszerek automatizálják a tranzakciókat és költséghatékony alternatívákat kínálnak. A tranzakciós szerepeken túl a kortárs POS-rendszerek interaktívak, lehetővé téve a vendéglátóiparban az ügyfelek számára, hogy elektronikusan rendeljenek, foglaljanak és rendezzék a számlákat. A POS-rendszerek előnyei nyilvánvalóak abban, hogy képesek racionalizálni a kiskereskedelmi műveleteket, automatizálni a tranzakciókat és nyomon követni az alapvető értékesítési adatokat. A modern POS-rendszerek alkalmazkodóképesek, programozhatók és kompatibilisek harmadik féltől származó szoftverekkel, lehetővé téve a kiskereskedők számára, hogy a funkciókat az egyedi igényekhez igazítsák. Ahogy a vállalkozások a versenyképességre törekszenek, a POS-rendszerek folyamatos innovációja alakítja a kiskereskedelmi piacot [3]

2.1.2. Square

A Square újradefiniálta a POS-t azáltal, hogy leegyszerűsítette a tranzakciókat minden méretű vállalkozás számára. Felhasználóbarát felülete zökkenőmentesen dolgozza fel a fizetéseket, és mind a boltokban, mind az online felületen megoldásokat kínál. Az olyan funkciókkal, mint a valós idejű leltárkezelés és a teljeskörű elemzések, a Square lehetővé teszi a vállalkozások számára, hogy hatékonyan alkalmazkodjanak a piaci trendekhez [4].

2.1.3. Shopify

Az e-kereskedelemben való jártasságáról ismert Shopify kiterjeszti képességeit a POS-ra is. Ez a platform zökkenőmentesen integrálja az online és offline értékesítést, teljeskörű megoldást nyújtva a vállalkozások számára. A Shopify POS funkciói közé tartozik a készletszinkronizálás, a többcsatornás értékesítés és az ügyfélkapcsolat-kezelés, így biztosítva a kereskedelem egységes megközelítését [5].

2.1.4. Lightspeed

A Lightspeed a kiskereskedelmi, és a vendéglátóipari szektor számára átfogó POS-megoldást kínál. Funkciói közé tartozik a készletgazdálkodás, az alkalmazottak nyomon követése és a széleskörű analitika. A Lightspeed kiemelkedik alkalmazkodóképességével, mivel személyre szabott megoldásokat kínál a legkülönbözőbb iparágak vállalkozásai számára [6].

2.1.5. LS Central

Az LS Central a Microsoft Dynamics 365 Business Central rendszert, a Microsoft vezető ERP-megoldását bővíti ki kiskereskedelmi funkciókkal. Egyetlen platformon kínálja az ügyfelek számára a pénzügyi funkciókat (a Microsoft ERP-t), az értékesítési pontot (POS), az üzletmenedzsmentet, az ellátási lánc kezelését, a készletgazdálkodást, az e-kereskedelmet és az ügyfélhűséget. [7]

2.1.6. Összefoglaló

A POS-rendszerek összehasonlítása látható az 1. ábrán.

Tulajdonságok	Square	Shopify	Lightspeed	LS Central
Készletkezelés	Valós idejű nyomonkövetés	Szinkronizáció különböző csatornák között	Átfogó eszközök kiskereskedelemhez és vendéglátáshoz	Üzleti centralizáció, mely egységesíti a készletkezelést, értékesítést és üzleti folyamatokat
E-kereskedelmi integráció	Zökkenőmentes online és offline tranzakciók	Többcsatornás értékesítés	Kiskereskedelemhez és vendéglátáshoz való integráció	Az online és offline értékesítés integrációja üzleti folyamatokkal
Analitika és jelentések	Bevásárlási trendek elemzése	Robusztus elemzések döntéshozatalhoz	Részletes jelentések és elemzések	Mélyreható analitikai eszközök az üzleti teljesítmény elemzéséhez és optimalizálásához
Rugalmasság és alkalmazkodóképesség	Megfelelő minden méretű vállalkozásnak	Skálázható különböző vállalkozásokhoz	Személyre szabott megoldások kiskereskedelemhez és vendéglátáshoz	Rugalmas és alkalmazkodó platform a vállalati igényekhez
Felhasználói felület	Felhasználóbarát és könnyen elérhető	Intuitív felület könnyű navigáláshoz	Felhasználóbarát ipar-specifikus megközelítés	Átlátható és könnyen kezelhető felhasználói felület, mely támogatja az üzleti folyamatokat

ábra 1: POS rendszerek összehasonlítása[8]

Mint az LS Retail ehf. alkalmazottja, kizárólag az LS Central-t választottam a felsorolt termékek közül. Az elsődleges függőségem középpontjában ez az alkalmazás áll, és a projekt célja az erre épülő német lokalizáció kiegészítése.

2.2. Microsoft Dynamics 365 Business Central

A Business Central egy olyan vállalatirányítási megoldás kis- és középvállalkozások számára, amely automatizálja és leegyszerűsíti az üzleti folyamatokat, segítve ezzel a vállalkozások mindennapjait. A rendkívül alkalmazkodóképes és funkciókban gazdag Business Central lehetővé teszi a vállalatok számára, hogy hatékonyan kezeljék üzleti tevékenységüket, beleértve a pénzügyeket, a gyártást, az értékesítést, a szállítást, a projektmenedzsmentet, a szolgáltatásokat. A vállalatok könnyen hozzáadhatnak olyan funkciókat, amelyek a működési régióra szabottak, és testre szabhatóak, hogy még a legspecifikusabb iparágakat is támogassák. A Business Central gyorsan bevezethető, könnyen konfigurálható, és az egyszerűség jellemzi az innovációt a terméktervezésben, fejlesztésben és a használhatóságában. [9]

2.3. Local Functionality for LS Central DE

Az LS Central német lokalizációja olyan funkcionalitást biztosít, amely megfelel a pénzügyi törvényi követelményeknek, különös tekintettel a TSE (Technical Security Element) használatával kapcsolatban. A TSE Németországban egy kötelező komponens, amelyet minden POS rendszernek használnia kell. Egy tanúsított biztonsági modult, biztonságos tárat és egy digitális interfészt foglal magában. Mind a POS, mind a TSE jelentése és regisztrációja kötelező az adóhatóságnál. Az LS Central integrálódik a Fiskaly API-val, amely egy felhőalapú köztes réteg a fiskaly CTSS/TSE számára. A fiskaly SIGN DE API megfelel a német KassenSichV (Kassensicherungsverordnung) előírásainak. Ez az integráció biztosítja a tranzakciók biztonságos kezelését és a pénzügyi előírások betartását [10].

2.4. Fiskaly

A fiskaly országspecifikus API-kat kínál a nyugta minden aspektusához, segítve a kiskereskedőket és a POS-szolgáltatókat az adózási szabályoknak való megfelelésben, teljesen felhő alapon [11].

2.4.1. KassenSichV

A KassenSichV a németországi pénztárgépek biztonsági előírásaira vonatkozó szabályozás, amely 2020. január 1-től lépett érvénybe. Az előírás célja a pénzügyi tranzakciók és adatok védelme, valamint az adóelkerülés elleni küzdelem. A szabályozás előírja, hogy minden elektronikus pénztári rendszernek rendelkeznie kell Technische Sicherheitseinrichtung (TSS)-el, amely garantálja a tranzakciók biztonságos rögzítését és a jogszabályi előírásoknak való megfelelést. Az TSS az adatok hitelesítését és biztonságos tárolását végzi, például digitális aláírások segítségével [12].

2.4.2. DSFinV-K

A DSFinV-K egy digitális interfész, amelyet a német adóhatóság számára hoztak létre, hogy a pénztári rendszerek által generált tranzakciós adatokat hatékonyan továbbítsák a hatósághoz. A szabályozás célja, hogy biztosítsa a pénzügyi tranzakciók átláthatóságát, és lehetővé tegye azok

automatikus ellenőrzését. A DSFinV-K segítségével a pénztári rendszerek digitálisan rögzítik és exportálják a szükséges adatokat, például a számlákat, a napi zárásokat és a tranzakciókat [13].

2.4.3. RKSV

Az RKSV az osztrák pénztári rendszerek szabályozása, amely előírja a pénztárgépek számára a megfelelő regisztrációt és a biztonságos adatkezelést. Az üzleteknek és vendéglátóhelyeknek alkalmazniuk kell olyan rendszereket, amelyek lehetővé teszik a tranzakciók digitális rögzítését és hitelesítését. Az osztrák adóhatóság számára biztosítani kell az automatikus és átlátható adatokat [14].

2.4.4. TicketBAI

A TicketBAI egy Spanyolországban alkalmazott rendszer, amely a pénzügyi tranzakciók és a számlák elektronikus rögzítését biztosítja. A célja, hogy megakadályozza az adóelkerülést és biztosítsa a tranzakciók megfelelő regisztrálását a hatóságok számára. A rendszer minden tranzakciót egyedi azonosítóval lát el, és azokat valós időben továbbítja az adóhatósághoz [15].

2.4.5. Vari*factu

A Vari*factu egy digitális számlázási rendszer, amely az adózási előírásoknak való megfelelés érdekében biztosítja a számlák és tranzakciók automatikus rögzítését és digitális aláírását. A rendszer által generált számlák megfelelnek a helyi jogszabályoknak, és biztosítják, hogy a tranzakciók megfelelő módon legyenek dokumentálva, hitelesítve és tárolva [15].

2.4.6. Fiskalcheck

A Fiskalcheck egy ellenőrző szolgáltatás, amely lehetővé teszi a pénztári rendszerek, különösen a pénzügyi rendszerek jogszabályi megfelelőségének tesztelését. A rendszer automatikusan ellenőrzi, hogy egy adott pénztári rendszer és az abban végrehajtott tranzakciók megfelelnek-e az aktuális adózási és biztonsági előírásoknak, mint például a KassenSichV és DSFinV-K [16].

2.4.7. Digital Receipt

A Digital Receipt a hagyományos papíralapú nyugták digitális változata. A digitális nyugta lehetővé teszi a vásárlók számára, hogy az üzletekben végzett tranzakcióik során elektronikus formában kapják meg a nyugtákat. Ez segít a környezetvédelmi követelmények betartásában, az adatok gyorsabb és hatékonyabb kezelésében, valamint segíthet az adózási szabályozásoknak való könnyebb megfelelésben [17].

2.5. Fiskaly API

A következő fejezetben megvizsgálom a fiskaly által biztosított releváns végpontokat.

2.5.1. Fiskaly SIGN DE V2

A fiskaly SIGN DE API egy RESTful típusú API, amely felhőalapú köztes réteget biztosít a fiskaly CTSS (Certified Technical Security System) / TSE (Technische Sicherheitseinrichtung) számára. Az API teljes mértékben megfelel a német KassenSichV (Kassensicherungsverordnung) előírásainak. A fiskaly SIGN DE API platformfüggetlen, és kizárólag szoftveres megoldást kínál. Az integrálásához mindössze egy stabil internetkapcsolatra van szükség [2].

2.5.2. Fiskaly DSFinV-K

A fiskaly DSFinV-K API egy RESTful API, amely egy pénztárázásra vonatkozó adatokat tárol, majd később exportálja azokat a DSFinV-K szerint. Az API által várt adatstruktúra nagyban hasonlít a DFKA Taxonomy 2.1.1-re, de összekapcsolja az adatokat a KassenSichV és a Management API bejegyzéseivel. A következő fejezetben részletesen kifejtem a szolgáltatás működését [18].

2.6. DSFinV-K végpontok

Az alábbiakban elemzem a szolgáltatás kapcsolódási pontjait üzleti logika alapján.

2.6.1. Authentication

Az API-hoz való hozzáférés csak érvényes JWT (JSON Web Token) / hitelesítési tokennel lehetséges, amelyet ezen végpont használatával lehet megszerezni. Az első kérésnél szükséges megadnunk az API key és API secret elemet, majd a sikeres válaszban megkapjuk a JWT-t és egy refresh token-t is. Amennyiben szeretnénk frissíthetjük a JWT érvényességét a refresh token segítségével [19].

2.6.2. VAT Definition

Az értékesítési adókulcsok összerendelése a törzsadatokban található pénzügyi adattárban van meghatározva. ÁFA-százalékok változása esetén nem szükséges új elemet hozzáadni. A beállítást bármikor elvégezhetjük és a leírást egyénileg módosíthatják [20].

Az API automatikusan biztosítja az 1-től 7-ig terjedő azonosítókat. Csak az 1000-tól származó vagy azt meghaladó azonosítókat lehet hozzáadni vagy megváltoztatni az API-n keresztül [20].

2.6.3. Purchaser Agency

Bizonyos vállalatok harmadik felek nevében gyűjtenek olyan bevételeket, amelyeket ügynöki bevételeknek neveznek. Ebben az esetben lehetőség van több ügynökség meghatározására és hivatkozására egy pénztárgépnél [21]. Ezt a funkciót nem fogjuk használni. Így a későbbiekben nem fejtem ki a működését.

2.6.4. Cash Register

A pénztárgépet az jellemzi, hogy képes mind az elektronikus rendelések fogadására, mind az elektronikus fizetési folyamatok lebonyolítására. Sok vállalatban olyan pénztárgépek vannak, amelyek rögzítik az üzleti tranzakciókat és elvégzik a fizetési tranzakciókat is. Azonban ha egyes pénztárgépeket (slave) központi pénztárgépen (master) keresztül zárnak le, ezt master-slave pénztárgép architektúrának nevezzük. Ebben az API-ban a master-slave kapcsolatot a megfelelő kliensazonosítón keresztül állítják be [22].

2.6.5. Cash Point Closing

A pénztárgép zárás az adott időszak összes egyedi tranzakcióját összegzi, amelynek típusa nyugta, és amelyek fontosak az ÁFA vagy jövedelemadó szempontjából. Ez azt jelenti, hogy csak azok a tranzakciók kerülnek figyelembevételre, amelyek a további adózási vagy pénzügyi feldolgozáshoz szükségesek. Minden összeget két tizedesjegy pontossággal jelenítünk meg.

A pénztárgép zárás lehetőséget ad arra, hogy matematikailag meghatározzuk a készpénzegenleget a pénztárban. A záró kimutatás egy összegzett, rendszerezett áttekintést nyújt az üzleti tranzakciókról, és a megfelelő pénztárgépen bármikor, akár minden nap is generálható [23].

2.6.6. Exports

Ez az exportálás CSV fájlok sorozatát biztosítja DSFinV-K szabvány szerint [24] A funkciót nem fogjuk implementálni, mivel kezdetben az exportot a fiskaly Dashboard-on keresztül indítjuk el, így a működését nem fejtem ki.

2.7. Verziókezelés

Ez egy olyan szoftvereszköz, amelyet a programozók a forráskód kezelésére használnak. Nyomon követi a forráskód módosításait, és segít a konfliktusok kezelésében [25].

2.7.1. Git

A Git egy ingyenes és nyílt forráskódú elosztott verziókezelő rendszer, amelyet úgy terveztek, hogy gyorsan és hatékonyan kezeljen kis és nagy projekteket egyaránt [26].

A Git és más VCS-ek (például a Subversion) közötti fő különbség abban rejlik, hogy hogyan kezeli az adatokat. Míg más rendszerek fájlalapú változások listájaként tárolják az információkat, a Git minden adatot ellenőrzőösszeggel lát el, mielőtt tárolná, és ezután az ellenőrzőösszeg alapján hivatkozik rá. Ez azt jelenti, hogy lehetetlen bármely fájl vagy könyvtár tartalmát megváltoztatni úgy, hogy a Git ne érzékelje azt. Ez a funkció a Git alapvető része, amely biztosítja, hogy ne vesszen el információ a szállítás során, és ne történjen fájlrongálás a Git észlelése nélkül [26].

A Git alapvető elemévé vált a mindennapi programozásnak, különösen a csapatmunka során, és széles körben elterjedt a szoftveriparban. A Git elsajátítása időt igényel a sok parancs miatt, de bizonyos parancsokat gyakran, ha nem is naponta használunk [27].

2.7.2. Bitbucket

A Bitbucket, az Atlassian tulajdonában lévő platform sokoldalú webes megoldás a forráskódok kezelésére és együttműködésére. A Git és a Mercurial támogatásával megkönnyíti a hatékony verziókezelést, a kódok közötti együttműködést és a folyamatos integrációt. A legfontosabb funkciók közé tartoznak az elágazási stratégiák, a beépített Jira integráció, a Bitbucket Pipelines a CI/CD-hez, valamint az olyan biztonsági intézkedések, mint a hozzáférés-szabályozás. Rugalmassága és fejlesztőbarát funkciói miatt népszerű választás a különböző projekteken dolgozó csapatok számára, a nyílt forráskódú kezdeményezésektől a vállalati szintű alkalmazásokig [28].

2.7.3. Azure DevOps

Az Azure DevOps a Microsoft által kínált fejlesztési eszközök és szolgáltatások átfogó készlete, amelynek célja a szoftverfejlesztési életciklus racionalizálása és fejlesztése. Integrált környezetet biztosít a verzióvezérléshez, a build-automatizáláshoz, a kiadáskezeléshez és az alkalmazásfigyeléshez. Az Azure DevOps olyan funkciókkal, mint az Azure Repos a verziókezeléshez, az Azure Pipelines a folyamatos integrációhoz és telepítéshez, az Azure Boards a feladatkövetéshez, valamint az Azure Test Plans a teszteléshez, hatékony eszközt kínál a csapatok számára a kiváló minőségű szoftverek tervezéséhez, fejlesztéséhez, teszteléséhez és szállításához. Felhőalapú jellege biztosítja az együttműködést és a hozzáférhetőséget, így hatékony megoldást jelent a modern alkalmazásfejlesztéssel foglalkozó csapatok számára [29].

2.7.4. GitHub

A GitHub egy webalapú platform, amely elsősorban a Git használatával történő verziókezelésre összpontosít. A fejlesztők számára együttműködési térként szolgál, ahol kódot tárolhatnak és ellenőrizhetnek, projekteket kezelhetnek és szoftvert fejleszthetnek. Erőssége az egyszerűségében és a közösségi kódolási funkciókban rejlik, elősegítve a fejlesztők közötti együttműködést világszerte. A felhasználóbarát kezelőfelületével támogatja a hatékony kód-együttműködést, a pull-kérelmek, problémák és konfliktusok révén. Emellett a GitHub Actions kiterjeszti képességeit a CI/CD területére, lehetővé téve a felhasználók számára, hogy munkafolyamatokat automatizáljanak közvetlenül a tárolókon belül. A GitHub széles körben elterjedt a nyílt forráskódú közösségben, és ismert élénk ökoszisztémájáról és integrációiról [30].

2.7.5. GitLab

A GitLab egy mindenre kiterjedő DevOps platform, amely egységes megoldást kínál a verziókezeléshez, a folyamatos integrációhoz és szállításához (CI/CD), a hibakövetéshez, a kód felülvizsgálatához és még sok másához. Az együttműködésre helyezi a hangsúlyt, és lehetővé teszi a csapatok számára a munkafolyamatok leegyszerűsítését és a termelékenység növelését azáltal, hogy egyetlen integrált környezetben átfogó eszközkészletet kínál. Támogatja a Git-tárolakat, és átfogó funkciókat biztosít a teljes szoftverfejlesztési életciklus kezeléséhez. A GitLab

rugalmasságáról, skálázhatóságáról és széleskörű testreszabási lehetőségeiről ismert, így különböző méretű csapatok számára is alkalmas [31].

2.7.6. Összefoglaló

A 2. ábrán látható a táblázatos összefoglaló.

Szoftver	Bitbucket	Azure DevOps	GitHub	GitLab
Fő funkcionalitás	Version control és együttműködés Git tárolókkal	Kiterjedt DevOps platform verziókezeléssel, CI/CD-vel és még többel	Version control és együttműködés Git tárolókkal	Átfogó DevOps platform verziókezeléssel, CI/CD-vel és még többel
Verziókezelés	Git és Mercurial támogatás	Git és TFVC támogatás	Git	Git
CI/CD	Beépített Pipelines	Erős CI/CD YAML konfigurációval	GitHub Actions a CI/CD	Integrált CI/CD jelentős testreszabhatósággal
Integráció	Jó együttműködés Jira-val és más Atlassian eszközökkel	Integrálódik az Azure szolgáltatásokkal és harmadik féltől származó eszközökkel	Széleskörű integrációk és élénk ökoszisztéma	Széleskörű integrációs lehetőségek és beépített funkciók
Skálázhatóság	Skálázható különböző méretű csapatok számára	Megfelelő nagyvállalatok számára	Megfelelő különböző méretű projektekhez	Skálázható rugalmas telepítési lehetőségekkel
Árak	Az árak a felhasználók és a tárolók tárolókapacitásán alapulnak	Az árak a felhasználók és a használt szolgáltatások alapján változnak	Az árak a felhasználók és a privát tárolók használata alapján alakulnak	Az árak a felhasználók és a használt funkciók számának alapján alakulnak
Közösség és Támogatás	Jelentős előnyöket nyújt az Atlassian ökoszisztéma és	Kiterjedt dokumentáció és támogatási források	Széles és aktív közösség, elérhető a GitHub támogatás	Aktív közösség, a GitLab Inc. biztosítja a támogatást

ábra 2: Verziókezelő szoftverek összehasonlítása[8]

Mivel az Azure DevOps-ot céges szinten használjuk és a munkafolyamatainkat ez az eszköz támogatja, így én is ezt alkalmazom a munkám során.

2.8. Parancssor

Windows rendszeren két fő konzol áll rendelkezésre: a régebbi CMD.exe, amely a DOS parancsokat szimulálja, és a fejlettebb PowerShell, amely a Windows Vista óta elérhető, és kibővített adminisztrációs lehetőségeket kínál. A Parancssor a Windows verziójától függően a Start menü különböző helyein található, de a leggyorsabb, ha a futtatás ablakból indítjuk el [32].

2.9. PowerShell

A PowerShell egy keresztplatformos feladatautomatizálási megoldás, amely parancssori felületből, script-nyelvből és konfigurációkezelési keretrendszerből áll. A PowerShell Windows, Linux és macOS rendszereken is futtatható [33].

2.10. Powershell ISE

A Windows PowerShell Integrated Scripting Environment (ISE) egy grafikus felhasználói felületet biztosító host alkalmazás a Windows PowerShell számára, amely lehetővé teszi parancsok futtatását, scriptek írását, tesztelését és hibakeresését. Az ISE számos hasznos funkciót kínál, például több soros szerkesztést, tabulátoros kiegészítést, szintaxis kiemelést, szelektív végrehajtást, kontextusfüggő súgót és a jobbról-balra író nyelvek támogatását. A menüelemek és billentyűparancsok számos olyan műveletet megkönnyítenek, amelyeket a Windows PowerShell konzolban is elvégezhetünk. Például scriptek hibakeresésekor az ISE-ben egyszerűen, jobb kattintással állíthatunk be töréspontokat a kódsorokon [34].

2.11. Business Central Administration Shell

A Business Central Administration Shell egy PowerShell-eszköz, amely cmdleteket biztosít a Business Central helyszíni telepítéseinek kezeléséhez. Segítségével egyszerűen kezelhetők a Business Central Server- és webszerver-példányok, adatbázisok, felhasználók és egyéb konfigurációk. Emellett támogatja a bővítmények kezelését is, például azok közzétételét, szinkronizálását, telepítését és frissítését [35].

2.12. Összegzés

A 3. ábrán összehasonlítom a fenti eszközöket.

Jellemző	PowerShell	PowerShell ISE	Business Central Administration Shell
Alapvető cél	Feladatok automatizálása és scriptelés	PowerShell szkriptek írása és hibakeresése grafikus felületen	Business Central helyszíni telepítések adminisztrációja
Platformok	Windows, Linux, macOS	Windows	Windows (Business Central)
Felhasználói felület	Parancssori	Grafikus, több ablakos	Parancssori
Szkriptek futtatása	Parancsok és szkriptek futtatása	Szkriptek futtatása, írása és hibakeresés	Szkriptek és parancsok futtatása Business Central konfigurálásához
Szintaxis kiemelés	Nincs (alapértelmezett)	Igen, szintaxis kiemelés	Nincs (alapértelmezett)
Többsoros szerkesztés	Nincs	Igen	Nincs
Automatikus kiegészítés	Igen	Igen	Igen
Feladatkezelés	Általános feladatok és rendszergazdai teendők	Szkriptek és kód hibakeresése	Business Central specifikus adminisztrációs feladatok
Bővítménykezelés	Korlátozott	Korlátozott	Bővítmények közzététele, telepítése és frissítése
Rendszerezszközök	Széleskörű rendszerezszközök és adminisztráció	Rendszerezszközök, de szűkebb kör	Kifejezetten a Business Central rendszerekhez

ábra 3: Parancssorkezelők összehasonlítása

A PowerShell ISE-re esett a választásom, mivel a munkám során már létrehoztam scripteket amivel a szükséges műveleteket el tudom végezni.

2.13. Visual Studio Code

A Visual Studio Code alapvető eszközként szolgál a Business Central fejlesztésében, mivel egy könnyű, de mégis erőteljes forráskódszerkesztőt kínál, amely zökkenőmentesen működik Windows, macOS és Linux rendszereken. Ez az egyetlen szerkesztő, amely támogatott a Business Central bővítmények fejlesztésére, más alternatívák nem elérhetőek. Ezáltal a Visual Studio Code elengedhetetlen a Business Central ökoszisztémában dolgozó fejlesztők számára. Az AL Language kiterjesztés szintén szükséges, mivel biztosítja a bővítményfejlesztéshez szükséges speciális szintaxist és eszközöket. A Visual Studio Code és az AL Language bővítmény együtt egy hatékony, áramvonalas környezetet biztosít, amely kifejezetten a Business Central fejlesztés egyedi igényeihez van optimalizálva [36], [37].

2.13.1. AL Language Extension for Microsoft Dynamics 365 Business Central

A Visual Studio Code telepítése után az AL Language bővítmény beszerzése szükséges a Marketplace-ről, mivel nélküle a kód nem fordítható le. Az AL Language kiterjesztés számos funkcióval segíti a fejlesztést, javítva a kódolási élményt és gyorsítva a munkafolyamatokat. Kihasználhatjuk a szerkesztés teljes támogatását, például az IntelliSense kényelmét, amely hatékony kódkiegészítést biztosít: a szintaxis-kiemelést, amely javítja a kód olvashatóságát és a formázási lehetőségeket, és segítenek az egységes megjelenítésben [37].

Az első AL bővítmény gyorsan inicializálható az "AL: Go!" parancs beírásával a parancspalettába. Az artifact-támogatás lehetővé teszi különféle elemek, például Codeunits, Control Addins, Pages, Page Customizations, Page Extensions, Profiles, Queries, Reports, Tables, Table Extensions és XmlPorts egyszerű definiálását. A "Find all references" funkcióval alapos kódelemzést végezhetünk, amely megkönnyíti az adott szimbólum összes előfordulásának listázását. Az inkrementális fordítás gyors hibakeresést tesz lehetővé a Problémák ablakban megjelenő diagnosztikával, támogatva a zökkenőmentes fejlesztést [37].

A natív HTTP- és JSON-típusok használatával könnyedén elérhetők az Azure-funkciók és más webes szolgáltatások. Az alkalmazás telepítése és hibakeresése is egyszerű: az aktuális fejlesztői környezetbe telepíthetjük az alkalmazást a Ctrl+F5 billentyűkombinációval, míg az F5 segítségével tesztelhetjük és továbbfejleszthetjük. A függőségeket az app.json konfigurációs fájlban kezelhetjük, az app.json és launch.json fájlok gyors generálását pedig automatikus eszközök segítik. A gyors alkalmazásfejlesztés (RAD) során a Ctrl+Alt+F5 billentyűkombinációval indíthatjuk el az alkalmazást, az Alt+F5 segítségével pedig hatékonyan hibakereshetünk, ezzel felgyorsítva a fejlesztési folyamatot [37].

2.13.2. Egyéb bővítmények a produktivitás növelésére

Eric Wauters (Waldo), a csomag készítője hangsúlyozza, hogy ezek az eszközök nélkülözhetetlenek a Microsoft Dynamics 365 Business Central hatékony fejlesztéséhez. Elegendő az „AL Extension Pack” csomagot telepíteni, hogy hozzáférjünk a lenti bővítményekhez [38].

A CRS AL nyelvi bővítmény a Visual Studio Code-ban kifejezetten a Microsoft Dynamics 365 Business Central fejlesztésére van szabva, és bővíti az AL nyelv támogatását új funkciókkal. Az AL Variable Helper segíti a fejlesztőket az AL-változókkal kapcsolatos információk megértésében, így elősegíti a hatékony és hibamentes kódolást. Az AL Code Outline funkció az AL-kód strukturált ábrázolását biztosítja, ami egyszerűsíti a navigációt és a kódszervezés megértését. Az AL Object Designer vizuális felületet kínál az AL objektumok tervezéséhez, megkönnyítve a különböző objektumok létrehozását és kezelését [39], [40], [41].

Az AL Toolbox hasznos parancsokat és kódrészleteket tartalmaz, amelyek növelik a fejlesztők termelékenységét és csökkentik a manuális munka szükségességét. A Create GUID eszközzel könnyen létrehozhatók globálisan egyedi azonosítók közvetlenül a Visual Studio Code-ban. A vscode-icons vizuálisan látványos ikonokat biztosít a fájlok és mappák számára a VS Code

explorerben, ezáltal javítva a navigációt. A Docker integráció zökkenőmentes kezelést tesz lehetővé a Docker-konténerekkel, így a fejlesztők könnyen dolgozhatnak velük. A GitLens továbbfejleszti a VS Code Git képességeit, részletes betekintést nyújtva a kódtörténetbe. [42], [43], [44], [45], [46].

A Git History lehetővé teszi a felhasználók számára a Git előzmények megtekintését és a változások lekövetését, míg a PowerShell közvetlen integrációja script-készítést biztosít. A snippet-creator egyszerűsíti a kódrészletek kezelését, lehetővé téve a gyakran használt kódminták gyors újrafelhasználását. A Rest Client segítségével a fejlesztők közvetlenül HTTP-kéréseket küldhetnek és tekinthetik meg a válaszokat a Visual Studio Code-ban, megkönnyítve az API fejlesztést. A TODO Highlight kiemeli a kódban található TODO megjegyzéseket, míg a TODO Tree a függőben lévő feladatokat fa nézetben jeleníti meg [47], [48], [49], [50], [51], [52].

Az XLIFF Sync támogatja az XLIFF fájlok szinkronizálását, biztosítva a nyelvi fordítások konzisztenciáját. Az AL Test Runner leegyszerűsíti az AL kód tesztelését, míg az AL CodeActions kódműveleteket és gyorsjavításokat biztosít a fejlesztéshez. Az AL Navigator vizuális navigátort ad az AL kódhoz, segítve a fejlesztők munkáját a kód könnyebb navigálásában. Az AL XML dokumentációs támogatás segíti az XML megjegyzések kezelését, míg az mdAL a Visual Studio Code dokumentációs képességeit bővíti az AL kód számára [53], [54], [55], [56], [57], [58].

A NAB AL Tools a Microsoft Dynamics 365 Business Centralhoz kifejlesztett fordításkezelő eszköz. Az AL Object ID Ninja pedig az objektumazonosítók hatékony kezelésében segít. A Partial Diff vizuálisan jeleníti meg a kódváltozásokat, segítve a fejlesztőket a módosítások megértésében és felülvizsgálatában. A BusinessCentral.LinterCop linting funkciókat kínál, segítve a kódolási szabványok betartását és a hibák felismerését. A Bracket Select lehetővé teszi a kódban található zárójelek közötti blokkok kijelölését, amely javítja a kód navigálását és manipulálását a Visual Studio Code-ban [59], [60], [61], [62], [63].

2.14. Fejlesztői környezet

A Microsoft Dynamics 365 Business Central két verziója közül választhatunk: a felhőalapút (Cloud) és az on-premises (helyben telepített) megoldást. Mindkét verzió hasonló alapfunkciókat kínál, de a telepítés módja és a hozzá kapcsolódó költségek, valamint a testreszabhatóság és a skálázhatóság tekintetében fontos különbségek vannak [64]. Ezen felül a Dynamics 365 Business Central egy konténer alapú környezettel is rendelkezik [65].

2.14.1. SaaS Sandbox

A Business Central SaaS (Software as a Service) Sandbox egy dedikált környezetet biztosít a fejlesztők számára a dinamikus és biztonságos fejlesztési tevékenységekhez. A SaaS sandboxot választva a fejlesztéshez a felhasználók egy felhőalapú platformot használhatnak, amely számos, a Microsoft Dynamics 365 Business Central-ra szabott funkciót kínál. Ez a környezet megkönnyíti a zökkenőmentes fejlesztést, tesztelést és kísérletezést, lehetővé téve a fejlesztők számára bővítmények létrehozását, testreszabását és optimalizálását egy skálázható és stabil

keretrendszerben. A SaaS sandbox egy olyan ellenőrzött térként szolgál, ahol a fejlesztők hatékonyan építhetik és finomíthatják megoldásaikat a Business Central felhőalapú architektúrájával összhangban [66].

A Dynamics 365 Business Central Partner Sandbox licenc, amely kizárólag az érvényes Microsoft Partner Network (MPN) azonosítóval rendelkező partnerek számára készült, megkönnyíti a dedikált homokozók (Sandboxok) létrehozását tanulási, tesztelési és ügyféldemók céljából. A 2022 februárjában bevezetett licenc hozzáférést biztosít a Business Central Premium funkciókhoz, és csak a partner tenant-okon használható [67]. A Partner Sandbox licencek a Web Directen keresztül szerezhetők be, ami rugalmas, költséghatékony megoldásokat tesz lehetővé lejárat nélkül. A partnerek a licencet a experience.dynamics.com oldalon igényelhetik érvényes MPN azonosítóval, és jóváhagyás után megkapják az SKU-vásárláshoz szükséges tokent. A fizetési lehetőségek között szerepel a hitelkártya vagy számla az 500 dollárt meghaladó havi számlázás esetén [67].

A sandbox-környezet másolásakor óvintézkedéseket tesznek, többek között leállítják a feladatokat a munkasorban, törlik az integrációs beállításokat, és eltávolítanak bizonyos bővítményeket. A kimenő HTTP-hívások kezdetben blokkolva vannak, és jóváhagyást igényelnek. Bizonyos beállítások letiltásra kerülnek, és az adatokat biztonsági és adatvédelmi okokból törlik. Az első féltől származó bővítmények és lokalizációk kiigazításokon esnek át. A környezeti beállítások, beleértve az Application Insights kapcsolati karakterláncokat is, átmásolódnak a célkörnyezetbe, telemetriai lehetőségeket kínálva a felügyelethez [67].

A rendszergazdák további környezeteket hozhatnak létre a Business Central felügyeleti központban. A környezetek bármely országban vagy régióban létrehozhatók és használhatók, ahol a Business Central elérhető, beleértve azokat az országokat vagy régiókat is, ahol az ügyfél meglévő környezetei futnak. A környezetek kvótája akkor kerül alkalmazásra, amikor új környezetet próbál létrehozni vagy meglévő környezetet másolni a Business Central felügyeleti központban [68].

Nem támogatott a Visual Studio Code-ból olyan kiterjesztés közzététele, amelynek azonosítói megegyeznek az AppSource-on már közzétett kiterjesztéssel. Az azonosítók közé tartozik az appId és a verzió vagy a név, a kiadó és a verzió kombinációja. Ha mégis közzétesznek egy ilyen bővítményt, az bármikor eltávolítható [69].

2.14.2. Docker konténer

A Docker konténerek lazán elszigetelt környezeteket vezetnek be az alkalmazások csomagolására és futtatására. Ezek a konténerek fokozott elszigeteltséget és biztonságot nyújtanak, lehetővé téve számos konténer egyidejű működését egyetlen állomáson. A könnyű és önálló konténerek tartalmazzák az alkalmazás futtatásához szükséges összes szükséges elemet, kiküszöbölve a gazdatelepítésektől való függőséget. A Docker átfogó platformot kínál a konténerek életciklusának kezeléséhez [70].

A Dynamics 365 Business Central a Dockerrel felszerelt Windows rendszereken történő futtatásra tervezett artifact-ot kínál. A Business Central Dockeren történő futtatása a szokásos Docker-parancsokkal vagy a BCCContainerHelper PowerShell modul segítségével valósítható meg, amely jelentősen leegyszerűsíti a Docker-futtatási folyamatot [71].

A BCCContainerHelper modul létrehoz egy bcartifacts.cache nevű gyorsítótárat a C:\ meghajtón a artifact-ok hatékony tárolásához. Emellett létrehoz egy BCCContainerHelper nevű könyvtárat a C:\ProgramData alatt, amely az összes működési fájl helyétül szolgál. Ez a mappa, C:\ProgramData\BCCContainerHelper, a zökkenőmentes fájlátvitel és egyéb interakciók érdekében közös használatban van a konténerrel. Ha nincs megadva felhasználónév és jelszó, a modul jelszót kér, és az aktuális Windows felhasználónevet használja. Alternatív megoldásként, ha Windows-jelszót ad meg, a konténer beállítása integrálja a Windows-hitelesítést az állomással. A BCCContainerHelper asztali parancsikonokat hoz létre a Business Central webklienshez [72].

2.14.3. Helyi telepítés

A Business Central helyi telepítése azt jelenti, hogy a szoftvert egy dedikált környezetben, az adott szerveren vagy gépen futtatjuk, ezáltal biztosítva egy ellenőrzött helyet a fejlesztési, tesztelési és bemutató tevékenységekhez. A telepítés megkezdése előtt szükséges meggyőződni arról, hogy a rendszer megfelel-e a Microsoft által az adott Business Central verzióhoz meghatározott előfeltételeknek. Ezek általában magukban foglalják a kompatibilis operációs rendszert, az adatbázis-kiszolgálót, valamint egyéb szoftverkövetelményeket, például a .NET Framework szükséges verzióját [73].

A telepítési folyamat azzal kezdődik, hogy letöltjük a telepítőfájlokat a Microsoft hivatalos weboldaláról. A telepítési varázsló végig vezeti a felhasználót a folyamaton, lehetőséget nyújtva a különféle beállítások testreszabására. A folyamat során meg kell adnunk kulcsfontosságú paramétereket, például a kiszolgáló URL-címét, a portkonfigurációt és a hitelesítési módszert, igazodva a projekt igényeihez [73].

A helyi telepítés jelentős előnyökkel jár, mivel közvetlen ellenőrzést biztosít a környezet felett, és ideális fejlesztési, testreszabási, valamint alapos tesztelési tevékenységekhez. Ez lehetővé teszi a fejlesztők számára, hogy egy elszigetelt térben dolgozzanak, ahol a módosításokat biztonságosan kipróbálhatják és finomhangolhatják a projekt követelményeinek megfelelően. Emellett a helyi telepítés homokozós környezetet biztosít, amelyben különféle konfigurációk és funkciók tesztelhetők anélkül, hogy kihatással lennének az éles rendszerekre [73], [74].

A Business Central rendszer topológiája és komponensei a szoftver architektúrájának strukturált elrendezését jelentik. Ez magában foglalja a kiszolgálók, szolgáltatások és adatbázisok egymáshoz való viszonyát, amely alapvető a rendszer hatékony működése szempontjából. Az SQL adatbázis a Business Centralban az alkalmazási objektumok és üzleti adatok tárolását szolgálja, legyen az SQL Server, Azure SQL Managed Instance vagy Azure SQL Database. Több bérlős konfiguráció

esetén az alkalmazás- és üzleti adatok külön adatbázisokban tárolódnak, biztosítva a különálló bérleti adatok kezelését [75].

A Business Central Server a kommunikációs folyamatok központi eleme, amely a Windows Communication Frameworkre épül. Ez a .NET-alapú szolgáltatás kezeli az ügyfelek és az adatbázis közötti adatcserét, valamint olyan funkciókat, mint a hitelesítést, eseménynaplózást, ütemezett feladatokat és jelentéseket. A Web Server, amely az Internet Information Services (IIS) által működtetett Business Central Web Server komponenseket tartalmazza, lehetővé teszi a webes kliens és mobilalkalmazások hozzáférését [75].

A Business Central alkalmazás elérhető asztali számítógépeken, telefonokon és táblagépeken, dedikált platformként biztosítva a rendszerrel való interakciót. A rendszerben található webszolgáltatások, például a SOAP és az OData interfészek, megkönnyítik az alkalmazás funkcióinak külső rendszerekhez és felhasználókhöz való integrációját. Ezek a szolgáltatások rugalmasan konfigurálhatók és közzétehetőek, lehetővé téve például oldalak vagy kódegységek publikálását [75].

A Business Central Administration Shell PowerShell-moduljai révén a rendszergazdák hatékonyan kezelhetik a kiszolgáló és webes példányokat, adatbázisokat, felhasználókat, valamint bővítményeket. Ez a moduláris eszköz egyszerűsíti a telepítést és a karbantartást, támogatva az optimális környezetmenedzsmentet [75].

Ezen kívül a Business Central Server Administration eszköz egy Microsoft Management Console (MMC) formájában érhető el, amely a szerverek beállításának és teljesítményének optimalizálására szolgál. Felhasználóbarát felületén keresztül a rendszergazdák gyorsan testreszabhatják a konfigurációkat, biztosítva a telepítés egyedi követelményeinek megfelelő működést [75].

2.14.4. Összefoglaló

A környezetek összehasonlítása a 4. ábrán látható.

Környezet	SaaS homokozó	Helyi környezet	Docker környezet
Telepítési modell	Felhőben hostolt	Helyi telepítés	Containerizált, hordozható
Elérhetőség	Globálisan elérhető	Helyileg elérhető	Hordozható és elérhető
Erőforrásfelhasználás	Felhő erőforrásokra támaszkodik	Helyi erőforrásokat fogyaszt	Hatékony erőforrásfelhasználás
Izoláció	Elkülönített	Hatással van a helyi gépre	Izolációt biztosít
Konfigurációkezelés	SaaS szolgáltató által kezelt	Kézi beállítások szükségesek	A konfiguráció egyszerűsítése
Skálázhatóság	Skálázható igény szerint	Korlátozva a helyi erőforrások által	Skálázható és hordozható
Fejlesztési folyamat	Kiváló együttműködéshez	Ideális egyéni munkához	Konzisztens a környezetek között
Rugalmas felhasználás	Lehetnek korlátozásai	Nagyon rugalmas	Rugalmas egyensúlyt teremt

ábra 4: Környezetek összehasonlítása[8]

A fejlesztési és tesztelési folyamatok hatékonyságának növelése érdekében egy SaaS-alapú sandbox környezetet választottam. Ez a megoldás lehetővé teszi, hogy a fejlesztéseimet könnyedén végezzem, miközben a kollégáim menet közben tesztelhetik az elkészült funkciókat, és azonnali visszajelzést adhatnak. A valós idejű együttműködés révén gyorsan reagálhatok az észrevételekre, így biztosítva, hogy a fejlesztési folyamat rugalmasan és eredményesen haladjon.

Bár a SaaS környezet számos előnyt kínál, van egy jelentős korlátozása: nem támogatja a tesztalkalmazások telepítését. Ezért szükségessé vált egy helyi Business Central környezet létrehozása, amelyet kizárólag fejlesztésre és automata tesztek létrehozására használok.

Ez a kombinált megoldás biztosítja számomra a szükséges rugalmasságot, mivel a SaaS környezet az együttműködést és skálázhatóságot biztosít, míg a helyi környezet lehetővé teszi a tesztelési folyamatok mélyebb kezelését [76].

2.15. Tesztelés

Az Application Test Toolkit segítségével automatizáljuk és futtatjuk a teszteket, amelyeket írunk. A toolkit különböző funkciókat tartalmaz a tesztelés egyszerűsítésére és hatékonyabbá tételére. Ez magában foglalja a kódelemeket tesztfüggvényekkel, amelyek különböző alkalmazási területeket tesztelnek, valamint olyan kódelemeket, amelyek általános és alkalmazás-specifikus funkciókat tartalmaznak a tesztelési kód ismétlődésének csökkentése érdekében. Emellett az Application Test Toolkit tartalmaz alkalmazási objektumokat is, amelyek lehetővé teszik az alkalmazási tesztek futtatását [77].

Docker környezetben az Application Test Toolkit telepítése egyszerű. BCCContainerHelper segítségével futtatni kell az Import-TestToolkitToBCCContainer függvényt a -containerName paraméterrel a könyvtárak importálásához. Alternatív megoldásként, ha a New-BCCContainer függvényt használjuk a BCCContainerHelper PowerShell modulból a Docker-en való konténerek

létrehozásához, hozzáadhatjuk a `-includeTestToolkit` flag-et, ami automatikusan telepíti az Application Test Toolkitet a konténer létrehozásakor [72].

Helyi környezet esetében a függőségeket manuálisan tudjuk telepíteni a `Publish`, `Sync` és `Install` parancsokkal melyekhez importálnunk kell a `Microsoft.BusinessCentral.Apps.Management` modult [78], [79].

2.16. API fejlesztői platform

Az API platform egy olyan szoftverrendszer, amely integrált eszközöket és folyamatokat kínál, így hatékonyan készíthetünk, kezelhetünk API-kat (alkalmazásprogramozási interfészek) [80]. Az alábbiakban három API-tesztelésre és dokumentálásra használt eszközt hasonlítok össze, kiemelve azok célját, funkcióit és különleges jellemzőit.

2.16.1. Postman

A Postman egy átfogó eszköz az API-k fejlesztéséhez, teszteléséhez, automatizálásához és dokumentálásához. Könnyen kezelhető felületén egyszerűen hozhatók létre HTTP kérések és tesztelhetők API-k. Támogatja a környezetek kezelését, az automatizált tesztek végrehajtását, valamint a dokumentáció és monitorozás elkészítését. A rugalmasság, a gazdag funkcionalitás és a kiterjedt dokumentáció, ami kiemeli a sok hasonló eszköz közül [81].

2.16.2. Soap UI

A Soap UI elsősorban a SOAP és REST alapú webszolgáltatások tesztelésére szolgál, lehetővé téve a komplex szolgáltatások alapos vizsgálatát. Az eszköz támogatja a WSDL és Swagger dokumentumok importálását, valamint kiterjedt tesztjelentések generálását. Erősségei közé tartozik a célzott SOAP és REST támogatás, valamint az automatizálási és integrációs lehetőségek széles skálája [82].

2.16.3. Swagger UI

A Swagger UI az API-k dokumentálására és bemutatására szolgál egy interaktív webes felületen keresztül. Az alkalmazás lehetővé teszi a Swagger specifikációk generálását és exportálását, valamint az API végpontok kipróbálását közvetlenül a böngészőből. Kiemelkedő előnye az egyszerű használat és a gyors hozzáférés az API-dokumentációkhoz [83].

2.16.4. Összefoglaló

Az API fejlesztői platformok összehasonlítása az 5. ábrán látható.

Tulajdonságok	Postman	Soap UI	Swagger UI
Cél	API tesztelés, automatizálás, fejlesztés és dokumentáció	SOAP és REST webes szolgáltatások tesztelése	API dokumentálás és bemutatás interaktív webes felületen
Felhasználói felület	Könnyen kezelhető, lehetővé teszi az API tesztelését és HTTP kérések létrehozását	Átfogó és specifikus, lehetővé teszi a SOAP és REST szolgáltatások tesztelését, valamint a WSDL és Swagger dokumentumok importálását	Egyszerű és interaktív, lehetővé teszi az API leírásának megtekintését és a végpontok kipróbálását
Funkciók	Tesztelés, HTTP kérések szerkesztése, környezetek kezelése, automatizáció, monitorozás, dokumentáció készítés stb.	SOAP és REST API tesztelése, HTTP kérések szerkesztése, automatizáció, tesztjelentések generálása stb.	API dokumentáció megjelenítése, végpontok kipróbálása, Swagger specifikációk generálása stb.
Kiemelkedő jellemzők	Rugalmas automatizálási lehetőségek, környezetkezelés, kiterjedt támogatás és dokumentáció	Specifikus SOAP és REST támogatás, WSDL és Swagger importálási lehetőség, összetett tesztelési funkciók	Könnyen kezelhető felhasználói felület, interaktív végpont kipróbálási lehetőség, Swagger specifikációk exportálása

ábra 5: API fejlesztő platformok összehasonlítása[8]

Az API fejlesztéshez választott platformom a Postman, mivel széleskörű lehetőségeket kínál az API tesztelésére, automatizálására és dokumentálására. Intuitív felülete lehetővé teszi a HTTP kérések egyszerű létrehozását és tesztelését, ami különösen hasznos a gyors iterációkhoz és hibakereséshez. A Postman emellett erőteljes funkciókat is kínál, mint például környezetkezelés, automatizálás, monitorozás és beépített dokumentációs eszközök, amelyek egyszerűsítik az API fejlesztési folyamatokat [81]. A Postman rugalmas automatizálási lehetőségei és a széleskörű támogatásai számomra elengedhetetlenek, mivel ezek lehetővé teszik számomra, hogy hatékonyan fejlesszem és kezeljem az API-kat egy jól strukturált környezetben.

2.17. . Fejlesztési módszertanok

A fejlesztési módszertan egy rendszerezett megközelítés, amely segít a szoftverfejlesztési projekt irányításában és szervezésében. Célja a hatékony munkafolyamatok kialakítása, a feladatok hatékony végrehajtása és a minőségi szoftvertermék létrehozása. A módszertanok különböző elvek és gyakorlatok alkalmazásával segítik a projektcsoportokat a célkitűzéseik elérésében [84].

2.17.1. . Vizesés modell

A Vizesés modell a szoftverfejlesztés egy lineáris, folyamatábra-alapú módszertana. A folyamat lépéseit szigorúan előre meghatározzák és végrehajtják egymás után, anélkül, hogy visszalépések lennének. A fejlesztés lépései általában a következők: követelmények gyűjtése és elemzése,

tervezés, implementáció, tesztelés, telepítés és karbantartás. Az egyes fázisok közötti átmenet jellemzően szigorú és dokumentált. A módszertan előnye az egyszerűség és a könnyű követhetőség, de néha merevségéből adódóan nehezen alkalmazkodik változó követelményekhez vagy piaci változásokhoz [85].

2.17.2. Agilis módszertanok

Az Agilis módszertanok rugalmas és iteratív fejlesztési megközelítéseket alkalmaznak, amelyek az ügyfél igényeinek folyamatos visszajelzéseire és gyors változásaira reagálnak. Példák az agilis módszertanok közül a Scrum, a Kanban és az Extreme Programming (XP). A Scrum egy olyan keretrendszer, amely rövid, iteratív fejlesztési ciklusokat (sprinteket) használ, míg a Kanban egy folyamatos munkaáramlású módszertan, amely kiemeli és optimalizálja a munkafolyamatot. Az XP a minőségcentrikus fejlesztésre összpontosít, amely magában foglalja a folyamatos integrációt, a páros programozást és a tesztvezérelt fejlesztést. Az agilis módszertanok előnye az alkalmazkodóképesség, a rugalmasság és a gyors visszajelzés, amely lehetővé teszi a projekt résztvevőinek hatékonyabb együttműködését és a jobb minőségű szoftverek fejlesztését [86].

2.17.3. . Scrum

Scrum egy rugalmas és iteratív fejlesztési keretrendszer, amelyet gyakran alkalmaznak szoftverfejlesztési projektekben. A Scrum az agilis módszertanok közé tartozik, és arra összpontosít, hogy folyamatosan termeljen üzleti értéket a projekt során. A Scrum keretrendszer alapja a sprintek, amelyek rövid, általában két-három hetes időszakokra vannak tervezve. Minden sprint során a csapat kiválaszt egy adott munka mennyiséget, amelyet a sprint végére teljesíteni szeretne. A sprinteket rendszeres, rövid, napi állófogadásokkal (daily scrum) irányítják, amelyeken a csapattagok megosztják a munkájukat és az esetleges problémáikat. A Scrum fő szereplői a product owner, a scrum master és a fejlesztőcsapat. A product owner felelős az ügyfél igényeinek felméréseért és a termék backlog összeállításáért. A Scrum Master pedig a Scrum folyamat megértéséért, betartásáért és optimalizálásáért felel. A fejlesztőcsapat pedig az aktuális sprintben dolgozó szakemberekből áll, akik együttműködnek a sprint alatt azért, hogy a sprint végére teljesítsék a kitűzött célokat [87].

2.17.4. . Kanban

A Kanban egy folyamatos munkaáramlású módszertan, amely az ütemezett feladatok és a munkaáramlás optimalizálására összpontosít. A Kanban táblát használ, amelyen az egyes feladatok (kártyák) folyamatosan mozognak a különböző fázisok (oszlopok) között. Ezek az oszlopok általában reprezentálják a munka különböző stádiumait, például "Várakozás", "Folyamatban" és "Kész" [88].

A Kanban fő elve az, hogy korlátozza a munkában lévő feladatok számát az adott időszakban, ezáltal minimalizálva a túlterheltséget és az időpazarlást. A csapat rendszeres, folyamatosan összehangolt daily standup-okat tart, amelyeken átbeszélnek az aktuális feladatokat és az esetleges problémákat [88].

A Kanban alkalmazkodóképességet és könnyű bevezetést kínál, mivel nem igényel szigorú szabályokat vagy előre meghatározott iterációkat, mint például a Scrum. Ehelyett a Kanban folyamatosan optimalizálja a munkafolyamatot és a csapat teljesítményét, miközben rugalmasan alkalmazkodik a változó igényekhez és körülményekhez [88].

2.17.5. . Összefoglaló

A csapatban a Scrum alapelveit követjük. Sprintjeink időtartama két hét, az utolsó hét csütörtöki napján Grooming sessiont tartunk, ahol a még nem pontozott feladatokat megvizsgáljuk és megbecsüljük a ráfordítást. Pénteken a Sprint review-t a csapat egyik tagja tartja, ahol összegyűjti, hogy milyen feladatokat végeztünk el és egy rövid előadás keretein belül bemutatja.

A következő hétfői nap az új sprint első napja, ezt egy Retro Session-nel szoktuk kezdeni amiben felfedjük az előző sprintben felmerülő pozitív és negatív eseményeket. A felmerült problémákra a csapat minden tagja tud reflektálni. Ezen felül minden nap tartunk egy 10-15 perces Daily Scrum meetinget.

3. KÖVETELMÉNY SPECIFIKÁCIÓ

3.1. Előkövetelmény

- Regisztrált fiskaly fiók (<https://dashboard.fiskaly.com/>), beállított szervezet.
- API Key és API Secret létrehozása a szervezethez.
- Telepített Business Centrál DE kliens LS Central w1 és LS Central DE alkalmazásokkal.
- SIGN DE V2 beállítások végrehajtása.

3.2. Céges követelmény

- AppSource kompatibilis megoldás.
- Többnyelvűség biztosítása.
- Minden létrehozott kiegészítő mező illetve vezérlő automatikusan kap egy „LSCDE” prefixet, a továbbiakban ezt nem fogom kiemelni.
- Modul funkcionalitás szerint mappába szervezése.

3.3. Funkcionális követelmény

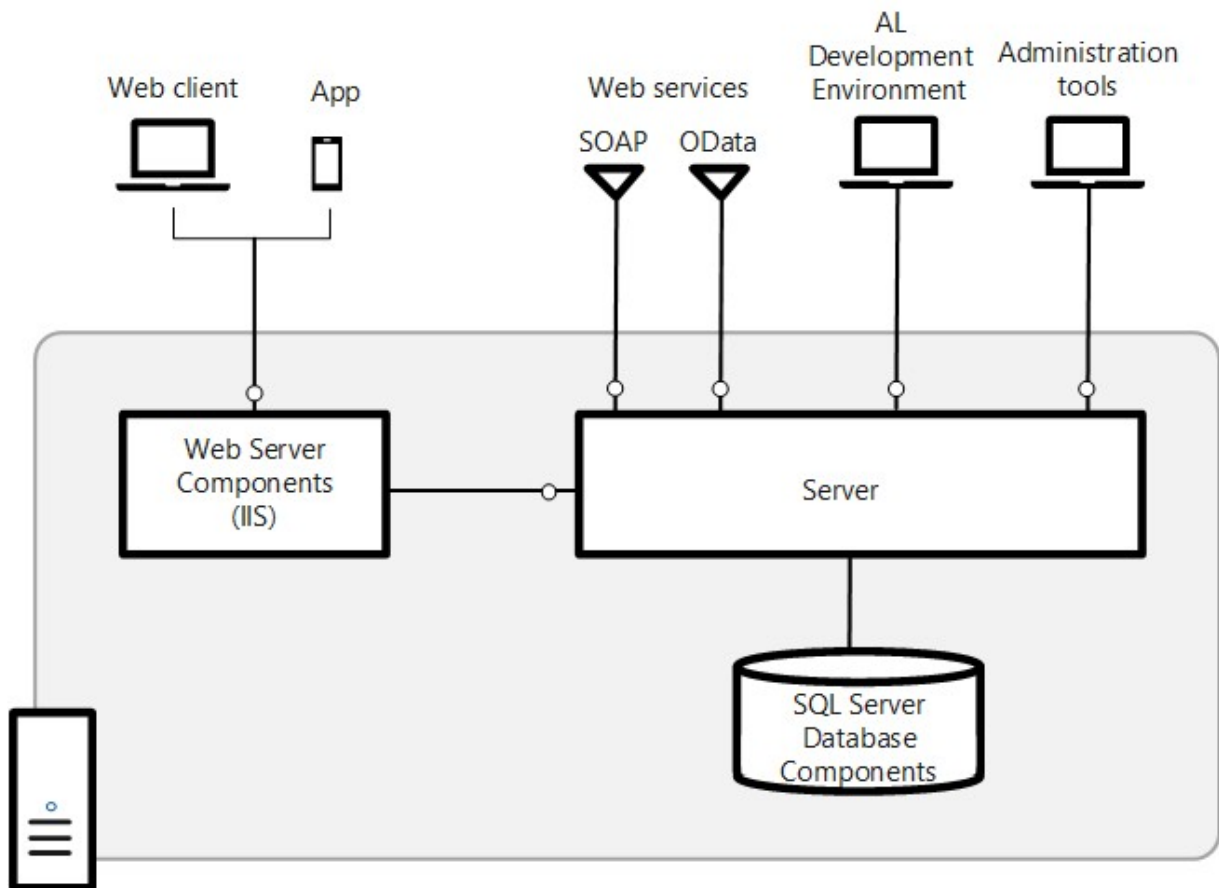
- Üzlet szinten kapcsolható működés.
- Fiskaly Setup rekordon beállítható kapcsolódási url.
- Minden rekord módosítás frissüljön a fiskaly rendszerben is.
- Fiskaly Áfa Definíciók létrehozása/módosítása/törlése Business Centralon keresztül. Naprakész adatok biztosítása.
- Cash Registerek létrehozása a terminálok alá, addicionális beállítások engedélyezése.
- A Cash Registerek nem törölhetők, de inaktíválni lehet őket.
- Brand, Model, Software Brand, Software Version és Base Currency Code mezőket lehessen frissíteni.
- Az első Cash Register mindig master (MASTER), a többi mindig az ehhez kapcsolódó slave (SLAVE_WITH_TSS).
- A master Cash Register zárja a hierarchiában alatta álló slave Cash Registereket.
- Amely üzletben a DSFinV-K engedélyezve van, ott minden tranzakció tartalmazza, hogy mely Master Cash Registerhez tartozik amennyiben a “Cash Tender (Fiskaly)” mező aktív a tranzakcióhoz tartozó “Tender Type Setup” rekordon.
- Pénztárzárások létrehozása a napi záróösszefoglaló elkészítésekor.
- Feladatvárólista létrehozása Pénztárzárás rekordok státuszának automatikus frissítésére.
- Pénztárzárás rekordok manuális státuszfrissítésének lehetősége.

4. TERVEZÉS

4.1. Rendszerterv

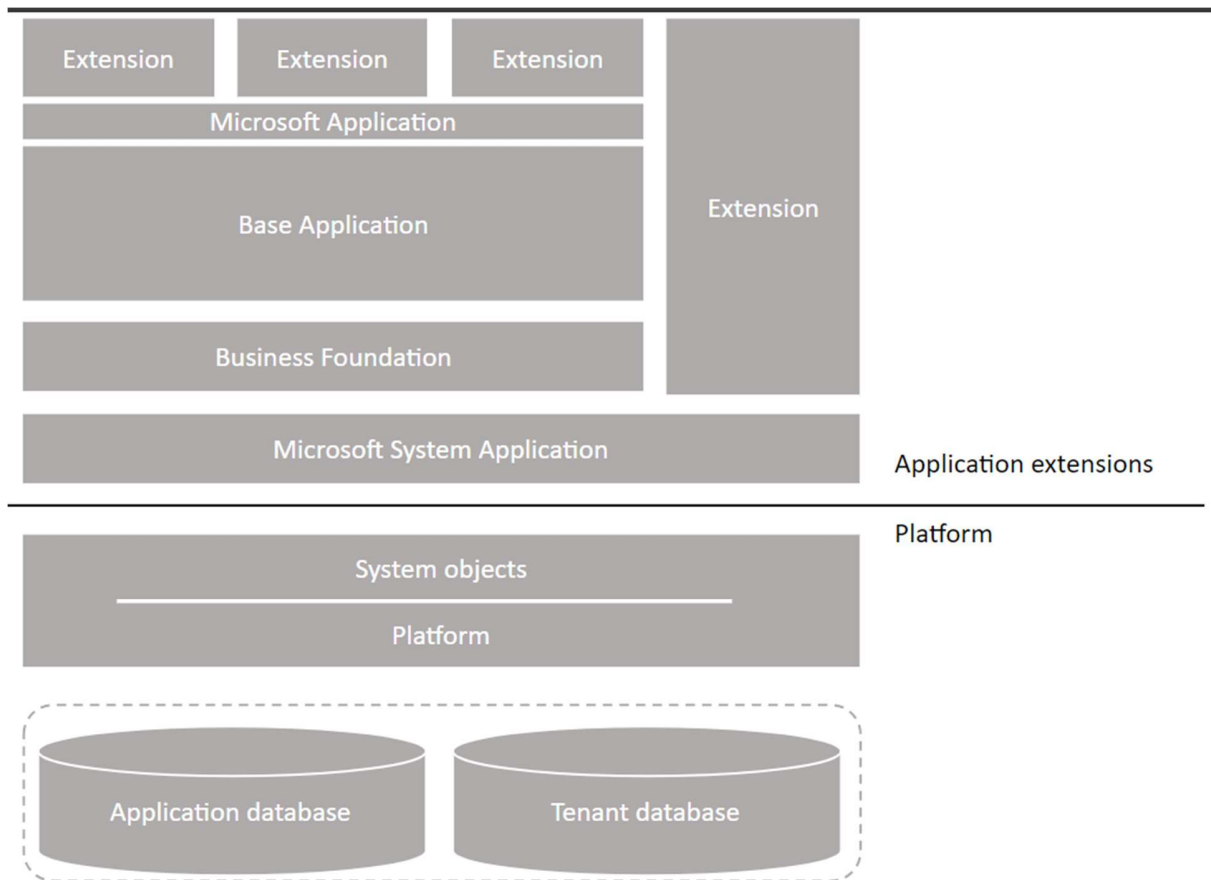
A megoldást a meglévő alkalmazásstruktúrába kell integrálnom.

A 6. ábrán látható az egyszerűsített Business Central rendszer topológia.



ábra 6: Rendszer topológia[75]

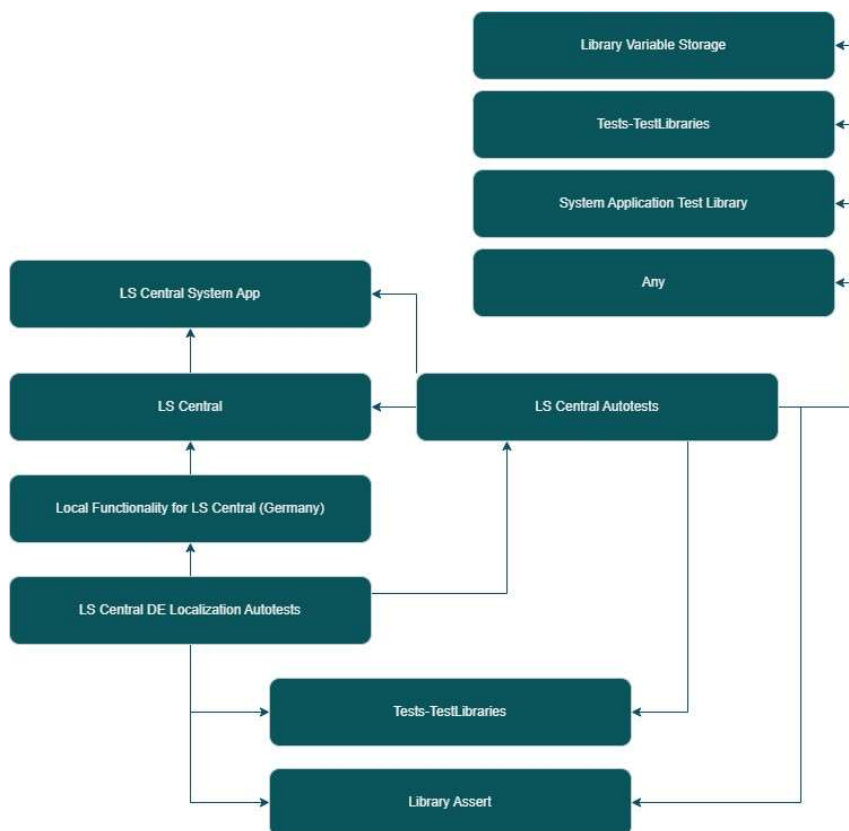
A Business Central alkalmazást két fő komponensre bonthatjuk fel, melyek az Application és Platform ahogy a 6. ábrán látható.



ábra 7: Business Central komponensek[89]

A saját bővítményeink automatikusan függenek platform és applikációs rétegtől, a továbbiakban ezt alapvetésként kezelem. Ezen beállításokat az app.json fájlban módosíthatjuk ahol megadható a minimum verzió mint függőség [90].

A 7. ábrán látható az egyszerűsített LS Central bővítmény és teszt-könyvtár hierarchia, melyben az Application és Platform függőségeket nem jelölöm.



ábra 8: LS Central alkalmazás hierarchia[8]

A Local Functionality for LS Central (Germany) és LS Central DE Localization Autotests alkalmazást fogom továbbfejleszteni.

4.2. Funkciólista

Az alábbiakban kifejtem, hogy milyen módon fogom megvalósítani a fejlesztést. A funkciók bemutatásakor alapvetésnek veszem, hogy a „<https://dsfinvk.fiskaly.com/api/v1>” megfelel a {{baseUrl}} változónak [18].

Csak azokat a funkciókat implementálom, ami szükséges az ügyféligény kielégítésére.

Minden operáció az LS Central felől indul, ezzel biztosítva az adatok integritását.

4.2.1. Autentikáció

Az authentication végpont várja az API key-t és API secret-et POST kéréssel, mely adatokat a fiskaly Dashboard-on keresztül korábban már inicializáltunk.

A következő struktúrát kell kialakítani és elküldeni erre a végpontra {{baseUrl}}/auth.

```

{
  "api_key": "{{fiskaly_api_key}}",
  "api_secret": "{{fiskaly_api_secret}}"
}
  
```

```
}
```

Küldés után visszakapjuk a következő választ, melyből minket egyedül az `access_token` fog érdekelni. Ez az érték tartalmazza a JWT, amire szükségünk van a többi folyamat futtatásához. A token idővel érvénytelenné válik, így kérhetünk újat, vagy a lejárat előtt frissíthetjük a `refresh_token`-nel az érvényességet [19].

```
{
  "access_token": "string",
  "access_token_claims": {
    "env": "TEST",
    "organization_id": "00000000-0000-0000-0000-000000000000"
  },
  "access_token_expires_in": 300,
  "access_token_expires_at": 1577833200,
  "refresh_token": "string",
  "refresh_token_expires_in": 300,
  "refresh_token_expires_at": 1577833200
}
```

4.2.2. VAT Definition rekord létrehozása/módosítása

PUT operációt kell végrehajtanunk a „`{{baseUrl}}/vat_definitions/vat_definition_export_id`” végpontra. Itt látható hogy query paramétert is alkalmazunk. Amennyiben a `vat_definition_export_id`-val már létezik rekord, akkor frissítjük, ellenkező esetben újat szúrunk be, ezt a mezőt szabadon tölthetjük egy integerrel. 1-7 ig beépített rekordok vannak és 999-től felfelé saját rekordokat tudunk beszúrni [91].

Szükséges megadnunk a következő két adatot:

```
{
  "percentage": "100",
  "description": "Test2"
}
```

Fontos, hogy törölni nem tudunk.

Ezt a funkciót a POS VAT Codes lista és Store karton oldalról kell elérnünk.

4.2.3. Cash Register létrehozása/módosítása

Két fajta pénztárat tudunk létrehozni, master-t és slave-et. A master Cash Register nem rendelkezik `master_client_id` tulajdonsággal és a `type` az MASTER. Mivel van SIGN DE V2 integrációnk így a `MasterCashRegisterSignDe` típusú pénztárakat készítünk [92].

A szükséges struktúra master esetén az alábbi módon néz ki:

```
{
  "cash_register_type": {
```

```

    "type": "MASTER",
    "tss_id": "00000000-0000-0000-0000-000000000000"
  },
  "brand": "string",
  "model": "string",
  "software": {
    "brand": "string"
  },
  "base_currency_code": "str"
}

```

Slave esetén pedig a json fájl a következő:

```

{
  "cash_register_type": {
    "type": "SLAVE_WITH_TSS",
    "master_client_id": "00000000-0000-0000-0000-000000000000",
    "tss_id": "00000000-0000-0000-0000-000000000000"
  },
  "brand": "string",
  "model": "string",
  "software": {
    "brand": "string"
  },
  "base_currency_code": "str"
}

```

Rekordot törölni itt sem tudunk, viszont lehet úgynevezett soft-delet-et alkalmazni, vagyis deaktiválhatjuk a létrehozott pénztárat [92].

A Cash Registereket a POS Terminal karton felületén egy aloldalon kell megjelenítenünk. Innen indítjuk a létrehozást/módosítást.

4.2.4. Cash Point Closing létrehozása

Létrehozáshoz PUT kérést kell küldenünk a `{{baseUrl}}/cash_point_closings/:closing_id` végpontra, a `closing_id` egy általunk megadott GUID érték. A meglévő tranzakciók alapján össze kell állítanunk a lenti hierarchiát [93].

```

{
  "client_id": "bf85e05d-35fa-4a7f-a415-35b82892f3bc",
  "cash_point_closing_export_id": 1,
  "head": {
    "first_transaction_export_id": "1",
    "last_transaction_export_id": "3",
    "export_creation_date": 1697029154,
    "business_date": "2023-10-11"
  },
  "transactions": [
    {
      "head": {

```

```

        "type": "Beleg",
        "storno": "false",
        "number": 1,
        "timestamp_start": 1697025583,
        "timestamp_end": 1697025583,
        "transaction_export_id": "1"
    },
    "data": {
        "full_amount_incl_vat": 30.00
    },
    "security": {
        "tss_tx_id": "355fbf74-8a58-432d-b8cc-88a60c2d5c19"
    }
}, {
    "head": {
        "type": "Beleg",
        "storno": "false",
        "number": 2,
        "timestamp_start": 1697026411,
        "timestamp_end": 1697026411,
        "transaction_export_id": "2"
    },
    "data": {
        "full_amount_incl_vat": 18.00
    },
    "security": {
        "tss_tx_id": "355FBF74-8A58-432D-B8CC-88A60C2D5C19"
    }
}, {
    "head": {
        "type": "Beleg",
        "storno": "true",
        "number": 3,
        "timestamp_start": 1697027749,
        "timestamp_end": 1697027749,
        "transaction_export_id": "3"
    },
    "data": {
        "full_amount_incl_vat": -30.00
    },
    "security": {
        "tss_tx_id": "355fbf74-8a58-432d-b8cc-88a60c2d5c19"
    }
}
]
}

```

A state mező lehetséges értékei: PENDING, WORKING, COMPLETED, CANCELLED, EXPIRED, DELETED és ERROR lehet [93].

Csak a COMPLETED státuszú Cash Point Closing-ok exportálhatóak [93].

4.2.5. Cash Point Closing lekérdezése

Mivel a feldolgozást a fiskaly végzi aszincron folyamatban, így a státusz frissítéséhez időnként be kell kérdeznünk a beküldött Cash Point Closing-ra. Ehhez egy GET kérést küldünk a `{{baseUrl}}/cash_point_closings/:closing_id` végpontra ahol a `closing_id` azonosítja a pénztárzárást [93].

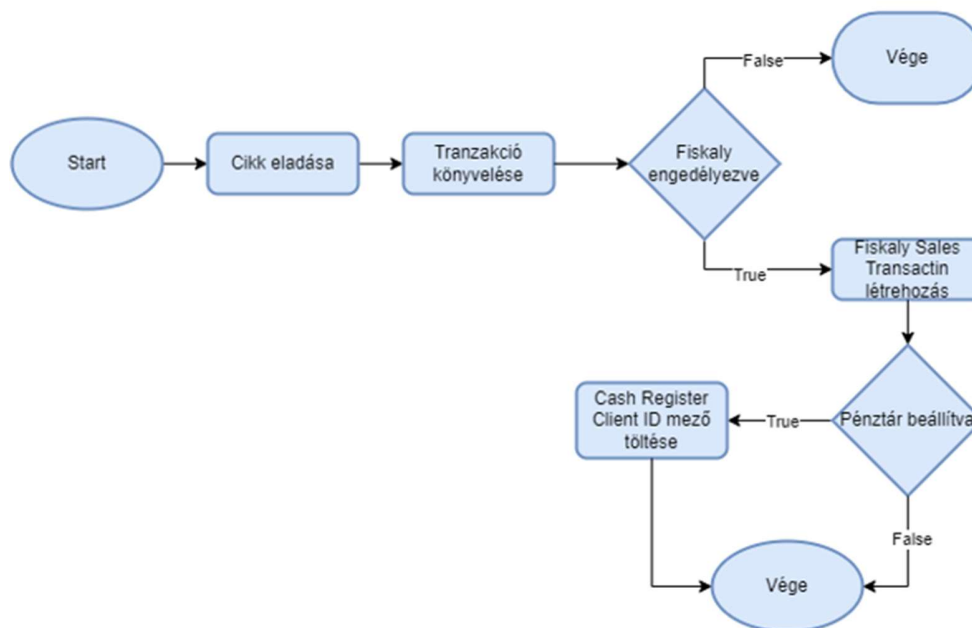
Lehetővé tesztek manuális, gombnyomásos és automatikus frissítést is feladatvárólista segítségével.

4.2.6. Cash Point Closing törlése

Törléshez a DELETE kérést használhatjuk a `{{baseUrl}}/cash_point_closings/:closing_id` végpontra. Ezzel a művelettel soft-delete történik, adatot nem veszítünk [93].

4.2.7. Fiskaly Sales Transaction kiegészítése

Minden eladás után amikor létrejön a Fiskaly Sales Transaction rekord, megvizsgáljuk, hogy az adott üzlet rendelkezik-e beállított pénztárral és ennek függvényében kitöltjük a „Cash Register Client ID” mezőt.



ábra 9:Tranzakció folyamat[8]

Amennyiben a napi zárást végezzük, megvizsgáljuk a folyamat elején az `OnBeforeStatementPost` eseményen, hogy a `DSFinV-K` engedélyezve van-e és az üzlet rendelkezik-e `MASTER` pénztárral, ha igen akkor inicializáljuk a Cash Point Closing rekordot, beállítjuk az első és utolsó tranzakció azonosítóját és egyéb szükséges adatokat.

Az esemény végén ellenőrizzük, hogy minden adat ki van-e töltve.

Ezek után megvárjuk, hogy a záróösszefoglaló nyilatkozat könyvelése a végéhez érjen, majd az OnAfterPostStatement eseményen amennyiben létezik Cash Point Closing rekord, beállítjuk a könyvelt nyilatkozat számát és elvégezzük a kommunikációt a fiskaly-val.

4.3. Fejlesztés előkészítése

Létrehozok egy Business Central DE környezetet a SaaS tenantunkon belül. A Users oldalon hozzáadom a csapat tagjait a környezethez, majd importálom a w1-es demo adatokat konfigurációs csomagok segítségével.

Készíték egy másolatot a jelenlegi konfigurált Cronus company-ról, melyen a fejlesztést végzem, illetve készíték egy másik másolatot is a QA tesztelésre. Így a rendszerben 3 company-m lesz, egy a demo adatokkal egy amiben a tesztelő kollégák dolgozhatnak és egy amin én fejlesztek.

A függőségeim telepítésére az install url-t tudom használni.

4.4. Tábla módosítások

Meglévő táblákat kiegészítünk új mezőkkel table extension segítségével, illetve új táblákat is létrehozunk.

4.4.1. Fiskaly Setup

Fiskaly Setup tábla a „Single-Record” (Setup) pattern szerint lett elkészítve, ide szükséges egy új mező „DSFinV-K Base Url”, melynek típusa Text[250]. Az alkalmazás telepítésekor lefutó InitSetup eljárás töltse fel ezt a mezőt <https://dsfinvk.fiskaly.com/api/v1> értékkel [94].

4.4.2. Store

A Store táblát kiegészítem a „Use DSFinV-K” mezővel. Az OnValidate triggeren ellenőrzöm, hogy a „Fiskaly Enabled” mező értéke igaz-e. Ezzel biztosítom, hogy csak akkor engedélyezhető a DSFinV-K használata, ha a Fiskaly integráció aktív.

A „Fiskaly Enabled” mező OnValidate triggerénél módosítást végzek, hogy amikor ennek a mezőnek az értéke false, a „Use DSFinV-K” mező értéke is automatikusan false legyen. Ezzel a beállítással elkerülhető, hogy a DSFinV-K funkció bekapcsolt állapotban maradjon, ha a Fiskaly funkciók le vannak tiltva.

4.4.3. POS VAT Code

A POS VAT Code táblában hozzáadom a „DSFinV-K Export ID” mezőt. Az OnValidate triggernél meghívom a ValidateDSFinVKExportID eljárást, amely ellenőrzi, hogy létezik-e már VAT Definition az adott ID-val. Ha igen, a rendszer hibát jelez, hogy elkerülje a duplikációkat.

4.4.4. Fiskaly Sales Transaction

Kiegészítem a Fiskaly Sales Transaction táblát a „Cash Register Client ID” mezővel, amely GUID típusú lesz, így a pénztárzárás során egyértelműen azonosítható az adott tranzakcióhoz kapcsolódó pénztár.

A tranzakciók állapotának jelölésére bevezetek egy új enum típusú mezőt, amely a következő értékeket tartalmazza: „, ”, ACTIVE, CANCELLED, FINISHED. Ezzel egyszerűbbé és áttekinthetőbbé válik a tranzakciók különböző állapotainak nyomon követése.

Szeretném rögzíteni azt is, hogy ki szakította meg a tranzakciót. Ennek érdekében létrehozok egy „Cancelled by User” nevű mezőt, amely a felhasználói azonosítót tárolja abban az esetben, ha megszakítás történik.

A teljesítmény növelése érdekében hozzáadok egy nem fürtözött (non-clustered) indexet a „FiskalyTransactionNumber” mezőhöz, ami a későbbiekben megkönnyíti a szűrést és gyorsabb adatlekérdezést tesz lehetővé.

4.4.5. Cash Register

A Cash Register (Pénztárgép) fejezet bemutatja a pénztárgép rendszerben használt kulcsokat, mezőket, valamint az ezekhez kapcsolódó eljárásokat.

Elsődleges kulcs

- POS Terminal No.: Ez a kulcs azonosítja az adott POS terminált, amelyhez a pénztárgép kapcsolódik.
- Line No.: A sorok azonosítására használt kulcs, amely az egyes pénztárgépeket különbözteti meg egymástól az adatbázisban.

A további szükséges mezőket létrehozom a DSFinV-K dokumentáció alapján.

4.4.6. Cash Point Closing

A Cash Point Closing (Pénztárzárás) fejezet bemutatja a pénztárzárás során használt kulcsot, a mezők szerkezetét, valamint a vonatkozó eljárásokat.

Elsődleges kulcs

- *Cash Point Closing Export ID*: Ez az egyedi azonosító kulcs szolgál a pénztárzárási adatok egyértelmű azonosítására és nyomon követésére az adatbázisban.

A további mezőket létrehozom a DSFinV-K dokumentáció alapján.

4.5. Felhasználói felület

A létrehozott új táblákat és mezőket meg kell jelenítenünk a felhasználói felületen is.

4.5.1. Fiskaly Setup

- “DSFinV-K Base Url” mező elhelyezése
- Cash Point Closings action nyissa meg a pénztárzárásokat

4.5.2. POS VAT Code

- “DSFinV-K Export ID” mező létrehozása
- “Generate VAT Definition” action frissítse a kapcsolódó fiskaly rekordokat

4.5.3. Store

- “Use DSFinV-K” mező elhelyezése
- “Updat DSFinV-K VAT Definitions” action frissítse a kapcsolódó fiskaly rekordokat

4.5.4. POS Terminal Card

- “Cash Registers” subpage jelenítse meg a kapcsolódó pénztárakat

4.5.5. Új oldalak

- Cash Registers
- Cash Register Card
- Cash Point Closings
- Cash Point Closing Card

4.6. Interfacek

Fontos, hogy a teszt projektben ki tudjuk kerülni a külső szolgáltatás hívását, így ahol lehetséges interface-t fogok használni.

- Authentication
- Cash Register
- Cash Point Closing
- Http request

4.7. Programmodulok

Szükséges feliratkoznom a LSC Statement-Post codeunit OnBefore és OnAfterStatementPost eseményére, hogy a funkció automatikusan tudjon futni amikor engedélyezve van. Erre egy handler codeunit-ot fogok létrehozni.

Az átláthatóság érdekében az általános eljárásokat a DsfinV-K Mgt. codeunit-ba helyezem. Minden egyéb specifikus funkcionalitás interfacen keresztül hívható, ennek megvalósítása érdekében egy SingleInstance codeunit-ot fogok készíteni, amely IoC konténerként funkcionál.

Mivel a Fiskaly Sales Transaction Fiskaly State mezője nem frissül automatikusan így erre a feladatvárólistába illeszthető codeunit-ot fogok létrehozni.

4.8. Fejlesztés tervezett menete

A következő sprintekbe szervezem a megvalósítást:

	43045	LSC DE DSFinV-K Authentication	 Ferenc Szilagyi		Closed	Description: https://dsfinvk.fiskaly.com/api/v1... https://developer.fiskaly.com/api/dsfinvk/v1#tag/Authentication/operation/authenticate... Discussion:
	43047	LSC DE DSFinV-K VAT Definition	 Ferenc Szilagyi		Closed	Description: https://developer.fiskaly.com/api/dsfinvk/v1#tag/VAT-Definition
	43048	LSC DE DSFinV-K Cash Register	 Ferenc Szilagyi		Closed	Description: https://developer.fiskaly.com/api/dsfinvk/v1#tag/Cash-Register
	43049	LSC DE DSFinV-K Cash Point Closing	 Ferenc Szilagyi		Closed	Description: https://developer.fiskaly.com/api/dsfinvk/v1#tag/Cash-Point-Closing
	43028	LSC DE DSFinV-K API	 Unassigned		Closed	Description: https://developer.fiskaly.com/api/dsfinvk/v1

ábra 10: User stories[8]

A fent tervezett sprinteknél a csúszás megengedett, amennyiben az üzleti logika vagy külső kommunikáció miatt komplikációk merülnek fel.

4.8.1. Analízis folyamat

- Regisztrálok egy fiókot, létrehozom a belépési kulcsokat.
- Megvizsgálom az endpointok-at, csatlakozom Postman segítségével és létrehozok adatot a fiskaly-ban.

4.8.2. Előkészítés

- Létrehozom a Business Central szerveret a SaaS környezetünkben.
- Publikálom az összes függőséget.

4.8.3. Meglévő táblák kiegészítése

- Az LS Central DE extension-ben létrehozom a mezőket.
- Az új mezőket megjelenítem a page-ken.

4.8.4. Cash Registers (pénztárak)

- Létrehozom a táblát és a hozzá tartozó lista illetve karton nézetet.

- CRUD operációk összekötése fiskaly-val.
- Subpage elhelyezése a meglévő POS Terminal page-n.

4.8.5. Cash Point Closing (pénztárzárás)

- Létrehozom a táblát és a hozzá tartozó lista illetve karton nézetet.
- CRUD operációk összekötése fiskaly-val.
- A listát a Fiskaly Setup page-ről egy action-ön keresztül tudjuk elérni.

4.8.6. DSFinV-K export

- Ellenőrzöm a beküldött adatok helyességét, módosításokat eszközölök amennyiben szükséges.
- Manuális státuszfrissítés action létrehozása
- Kapcsolatba lépek a fiskaly-val amennyiben valami nem egyértelmű

4.8.7. Feladatvárólista

- Létrehozom a programmodult ami naprakészen tartja a pénztárzárás rekordokat
- Tesztelem a programmodul működését

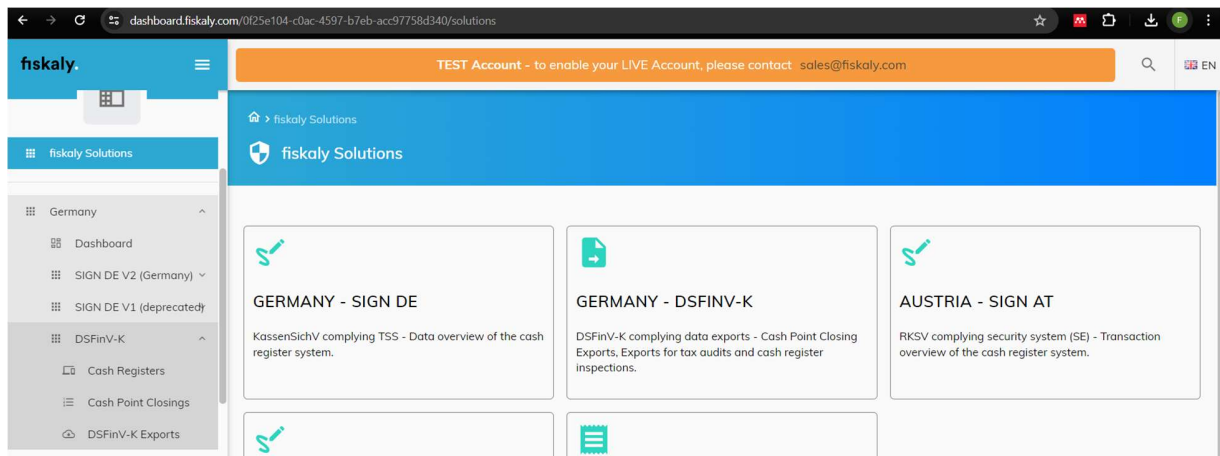
4.8.8. Tesztelés

- Létrehozom a Business Central szerveret lokálisan
- Függőségek publikálása
- Manuális tesztelés
- Tesztautomatizálás

5. FEJLESZTÉS

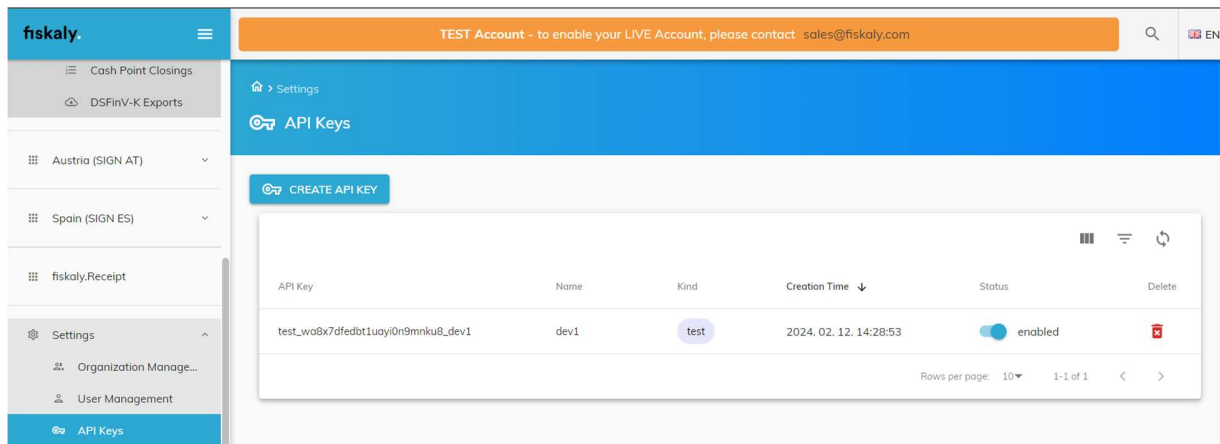
A fejlesztés megkezdése előtt frissítettem az összes eszközt, amit használni fogok a munkám során és elkészítettem a fiskaly fiókom.

Regisztrálás után a 11. ábrán látható felület tárult elém.



ábra 11: fiskaly dashboard[8]

A Settings fül alatt létrehoztam a csatlakozáshoz szükséges API Key-t és kimásoltam az API Secret-et. Fontos, hogy az API Secret-et csak a létrehozásnál látjuk, ha elfelejtjük elmenteni, akkor csak új kulcs létrehozásával tudjuk orvosolni a problémát. Ha sikeresen hozzáadtuk a kulcsot akkor a 12. ábrán látható módon jelenik meg.

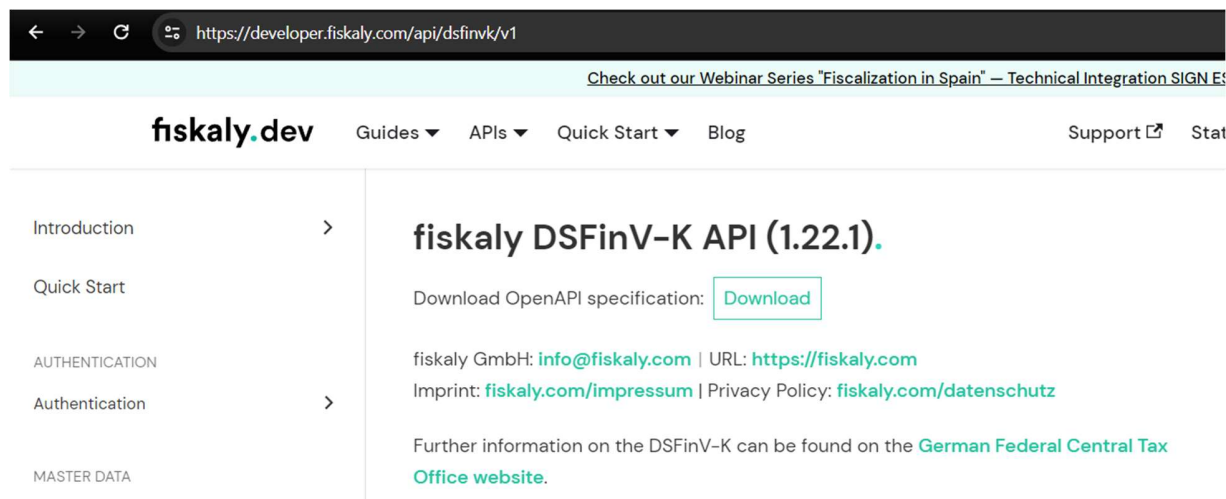


ábra 12: fiskaly API Kulcs[8]

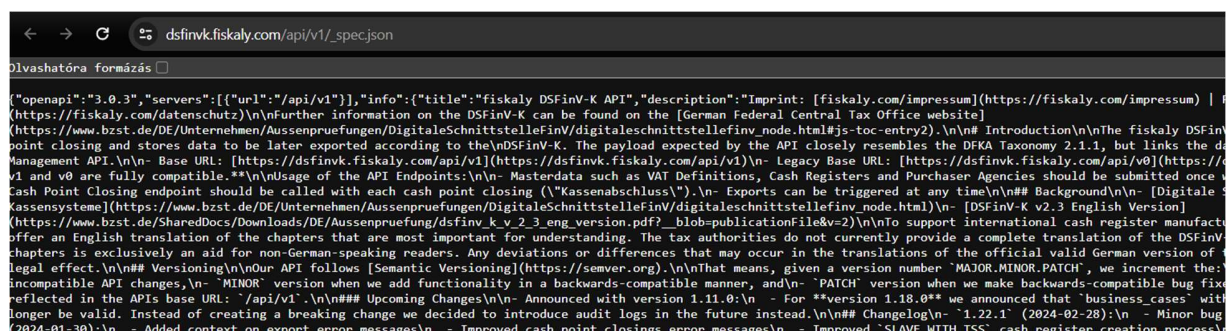
5.1. Analízis

Fontos tudnom, hogy a valóságban mely végpontok milyen adatokat várnak el, ezért Postman segítségével minden használni kívánt funkciót meg fogok vizsgálni.

A munkámat megkönnyítette a <https://developer.fiskaly.com/api/dsfinvk/v1> honlapon található OpenAPI definíció. A download gombra kattintva megnyílik egy új oldal és CTRL+S segítségével lementem a json kiterjesztésű fájlt.

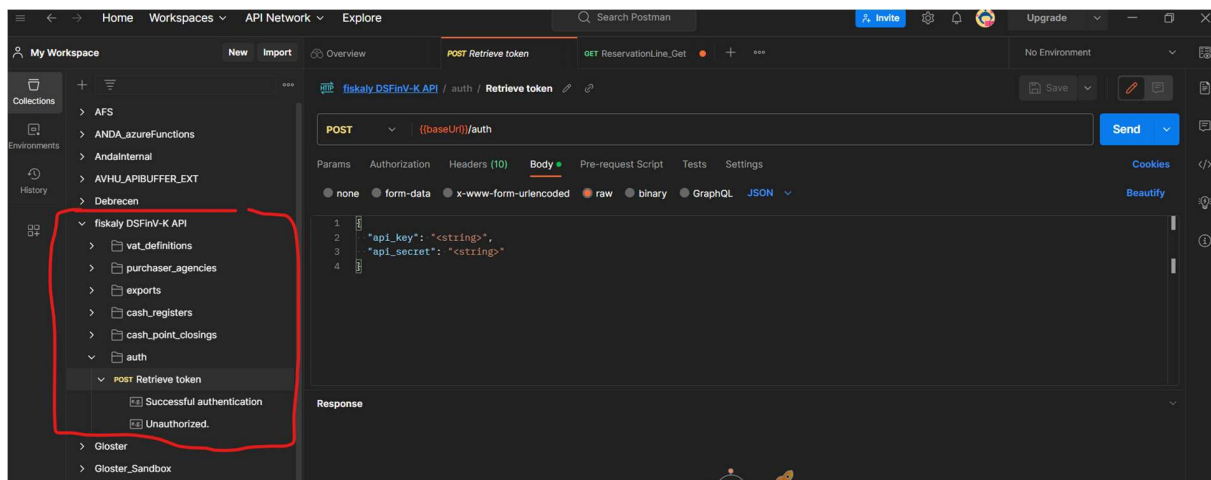


ábra 13: OpenAPI definíció[8]



ábra 14: OpenAPI definíció raw[8]

A letöltött állományt importálom Postman-be és előáll a 15. ábrán látható struktúra.



ábra 15: Postman gyűjtemény[8]

A végpontok tartalmaznak sikeres és sikertelen válaszokat is, így könnyen fel tudok készülni a hibák kezelésére.

5.2. Előkészítés

Ebben a szekcióban részletezem, milyen műveletek szükségesek a fejlesztés megkezdéséhez.

5.3. Fejlesztői környezet

Az adminisztrációs felületen az Environments fül alatt a new gomb használata után megjelenik egy varázsló, ahol beállítom a környezet nevét, típusát és országcódját. Az adatok megadása után, a create gomb elkészíti a környezetünket. Ami egy kis idő múlva aktív állapotra vált, ahogy a 16. ábrán megfigyelhető.

Environments > LSCDE_DSFinVK_Feri

Details				Active ⓘ
Name LSCDE_DSFinVK_Feri	Application Family Business Central	Country/region DE	Azure Region Germany	
Type Sandbox	Telemetry ⓘ Not Set (Define)	Security Group Not Set (Define)	Access with Microsoft 365 Licenses ⓘ Off (Modify)	

ábra 16: SaaS-környezet

Amint a környezet elkészült, installálom a szükséges bővítményeket. Az LS Central System és LS Central alkalmazást AppSource install url használatával fogom elvégezni. Ennek összeállítására PowerShell ISE scriptet használok amit a 17. ábrán láthatunk.

```
#####
$Url = "https://businesscentral.dynamics.com/+ $tenantID +/" + $env;
$installUrl = $Url + "?aid=FIN&page=2503&filter=%27ID%27%20IS%20%27" + $appid + "%27&signInRedirected=1"
Set-Clipboard -value $installUrl;
```

ábra 17: Appsource alkalmazások installálása[8]

A paraméterek beállítása után futtatom a scriptet és a generált url a vágólapra másolódik. Ezt a böngészőben megnyitom és végrehajtom az installálást mindkét alkalmazásnál.

A demo adatokat az LS Retail Portal weboldaláról töltöm le, ahol egy .zip fájlban található minden szükséges .rapidstart csomag, amelyek elegendőek a fejlesztésemhez. Ezt követően a Visual Studio Code segítségével publikálom a Local Functionality for LS Central DE alkalmazást. Végül frissítem a felhasználói listát, hogy kollégáim is hozzáférhessenek a környezetemhez.

5.4. Eljárások

A fejlesztés során az eljárásokat úgy strukturáltam, hogy paraméterként rekordot is kapnak, majd meghívnak egy overload-olt függvényt, ami csak paraméter értékeket várnak el. Erre azért van szükségem, hogy a tesztprojekt írásánál meg tudjam hívni az overload-olt függvényt csak paraméterekkel és ne legyenek a tesztéseim adatfüggők.

5.5. Logic Factory

Minden létrehozott vagy kiegészített táblán úgynevezett Logic Factory-t hozok létre, amit egy céges továbbképzésen sajátítottam el. A gyakorlatban ez azt jelenti, hogy minden table vagy table extension objektumban definiálok egy boolean típusú logicDefined változót és egy logic interface típusú változót.

Emellett létrehozok egy Define eljárást, amely egy interface-t kap paraméterként, hozzárendeli azt a globális logic változóhoz, majd a logicDefined változót igazra állítja.

Végül készítek egy Logic eljárást, amely interfészként tér vissza, és ellenőrzi, hogy a logicDefined értéke igaz-e. Ha igen, akkor a globális logic-kal tér vissza; ha nem, akkor az alapértelmezett implementációval. Ezzel a megoldással futásidőben tudom szabályozni, hogy egy adott rekord hogyan viselkedjen.

A Logic Factory struktúrája a 18. ábrán látható.

```

#region Logic factory
You, 5 months ago | 1 author (You)
var
    2 references
    _logic: Interface "LSCDE ICashPointClosing";
    2 references
    _logicDefined: Boolean;

11 references
local procedure Logic(): Interface "LSCDE ICashPointClosing"
    You, 5 months ago • Merged PR 29150: 52176 Epic de and other -Local...
var
    Default: Codeunit "LSCDE Cash Point Closing";
begin
    if not _logicDefined then
        Define(Default);
    exit(_logic);
end;

8 references
internal procedure Define(CashPointClosing: Interface "LSCDE ICashPointClosing")
begin
    _logic := CashPointClosing;
    _logicDefined := true;
end;

#endregion

```

ábra 18: Logic Factory struktúra[8]

5.6. Autentikáció

Szükséges a fiskaly DSFinV-K szolgáltatás használatához egy JWT, melyet API Secret és API Key segítségével kérhetünk. Az elvárt kimenetelünk tehát a JWT, ennek elérésére a következő interfaceket hoztam létre.

Az ITokenService implementációban az API Key és Secret segítségével kérem le a JWT-t, amely biztosítja az autentikációt és az API-hoz való hozzáférést.

```

Interface "LSCDE ITokenService"
  PropertyList
  GetToken(Store: Record "LSC Store") JWTToken: Text
  parameters
    [0] Store: Record "LSC Store"
  PostGetTokenRequest(ApiKey: Text; ApiSecret: Text) JWTToken: Text
  parameters
    [0] ApiKey: Text
    [0] ApiSecret: Text

```

ábra 19: ITokenService[8]

Ezután IAuthorize implementációban beállítottam a HttpHeaders Authorization értékét a kapott JWT-re ahogy a 18. ábrán látható.

```

Interface "LSCDE IAuthentication"
  Authorize(Store: Record "LSC Store", Headers: HttpHeaders)
  parameters
    [0] Store: Record "LSC Store"
    [0] Headers: HttpHeaders

```

ábra 20: IAuthentication[8]

Mindkét interfacet az Authentication programmodul valósítja meg.

5.7. Meglévő objektumok kiegészítése

Az alábbi entitásokhoz elkészítettem a table extension objektumot és bővítettem a mezőket illetve új eljárásokat adtam hozzájuk. Ezen felül bővítettem a hozzájuk tartozó felhasználói felületet is.

5.7.1. Fiskaly Setup

A Fiskaly Setup táblát kiegészítettem a DSFinV-K Base Url mezővel, amelynek Text[250] típust adtam, hogy a szolgáltatás címét tárolhassa. A meglévő InitSetup eljárást módosítottam, hogy ez a mező az alapértelmezett URL-lel töltődjön fel, így a rekord hozzáadásakor az ügyfélnek már nem szükséges manuálisan kitöltenie.

A felhasználói felületen a mezőt elhelyeztem a hozzá tartozó card page-en, ahol létrehoztam egy Cash Point Closings promoted action-t is. Ez az action megnyitja az újonnan létrehozott Cash Point Closings oldalt, amelynek létrehozását később részletezem.

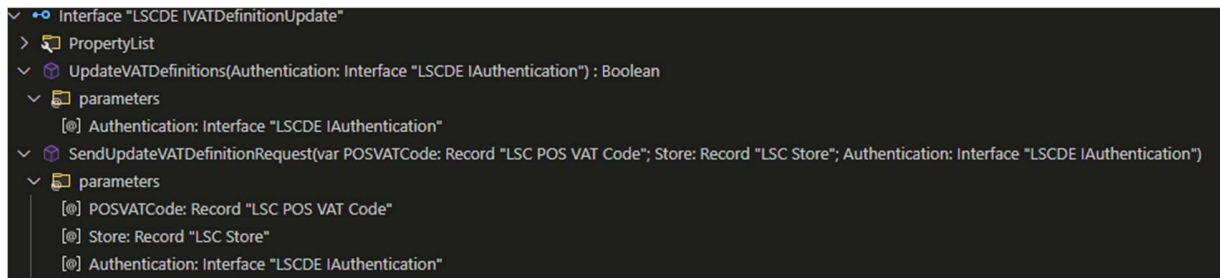
5.7.2. POS VAT Code

Az LS Central-ban megtalálható POS VAT Code táblát extendálom az új DSFinV-K Export ID integer típusú mezővel. Implementáltam a Logic Factory-t és létrehoztam az IPOSVATCode interface-t. Minden eljárást és trigger-t az interface implementációján keresztül hívok a 21. ábrán látható megkötésekkel.



ábra 21: IPOSVatCode[8]

A modify és insert trigger-en frissítjük a fiskaly VAT Definition listát. Erre az IVatDefinitionUpdate interface-t készítettem el és implementáltam a VatDefinitionUpdate codeunit-ban. Az IVatDefinitionUpdate interface látható a 22. ábrán.

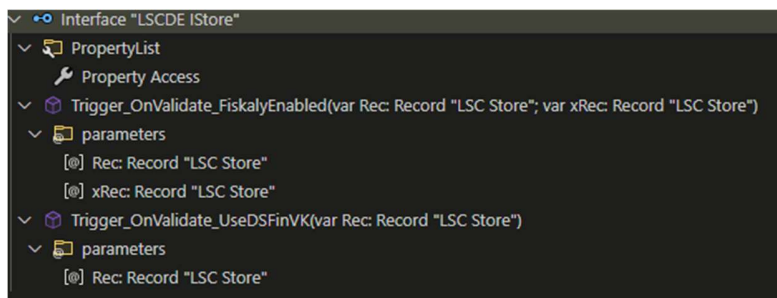


ábra 22: IVATDefinitionUpdate[8]

Bővítettem a felhasználói felületet az újonnan létrehozott mezővel és létrehoztam egy új action-t is, amellyel manuálisan tudjuk frissíteni a fiskaly VAT Definition listát.

5.7.3. Store

Bővítettem a Store LS Central táblát a Use DSFinV-K boolean mezővel, ahol a validate triggeren ellenőrzöm, hogy a Fiskaly Enabled mező értéke igaz-e, amennyiben nem, TestField hibát dobok. A rekordhoz tartozó interface struktúrája a 23. ábrán látható.



ábra 23: IStore[8]

5.7.4. POS Terminal Card

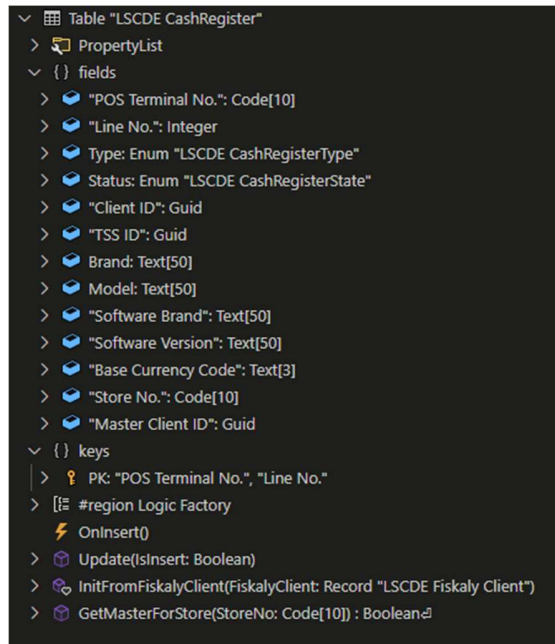
Létrehoztam a Cash Registers subpage-et melyet elérhetünk a POS Terminal kartonról. Ezen a felületen lehet létrehozni a pénztárakat.

5.8. Új objektumok

A következő szekcióban bemutatom az általam létrehozott összes új objektumot. Részletesen ismertetem a táblák mezőit és az azokhoz tartozó eljárásokat, valamint az új felhasználói felületet. Az eddigiekhez hasonlóan itt is alkalmaztam a Logic Factory mintát, így minden entitás interfésszel fog rendelkezni.

5.8.1. Cash Register

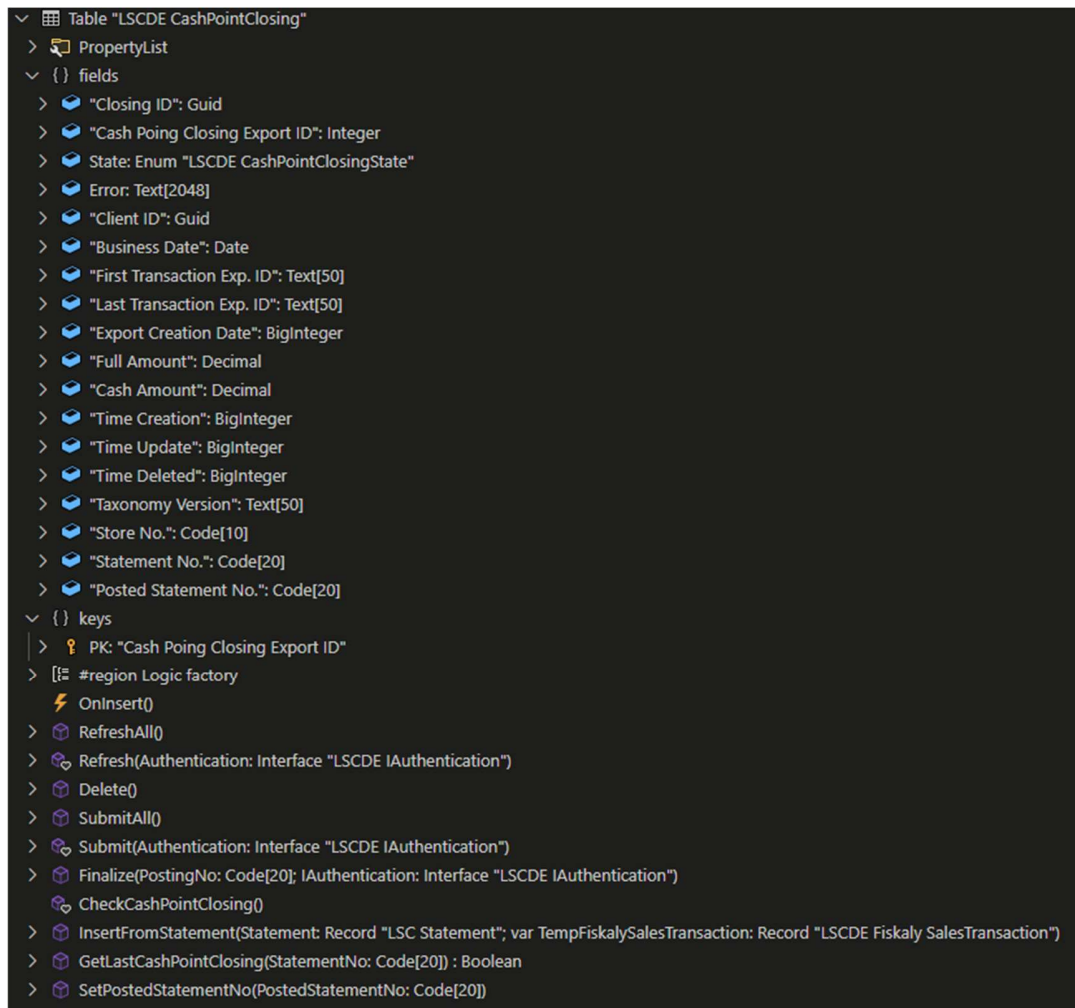
Központi entitásként készítettem el a Cash Register táblát, itt tárolok minden adatot a pénztárgépekről. A tábla szoros kapcsolatban van a Fiskaly Client-el, mivel a kliens alá tudunk pénztárgépeket rendelni. A tábla struktúrája a 24. ábrán látható.



ábra 24: Cash Register[8]

5.8.2. Cash Point Closing

A Cash Point Closing táblában tárolom a Statement Posting alatt összegyűjtött tranzakciók adatait. Tartalmazza az első és utolsó tranzakciót amit be kell vallani illetve kumulált összegeket is tárol.



ábra 25: Cash Point Closing[8]

5.9. Service Locator

A Service Locator Singleton codeunit-ot azzal a céllal készítettem, hogy a lazább kapcsolatot teremtsék a függőségek között. A Service Locator segítségével futásidőben tudom cserélni a szolgáltatás implementációját, ez a megközelítés segíteni fog a tesztalkamazás írásánál.

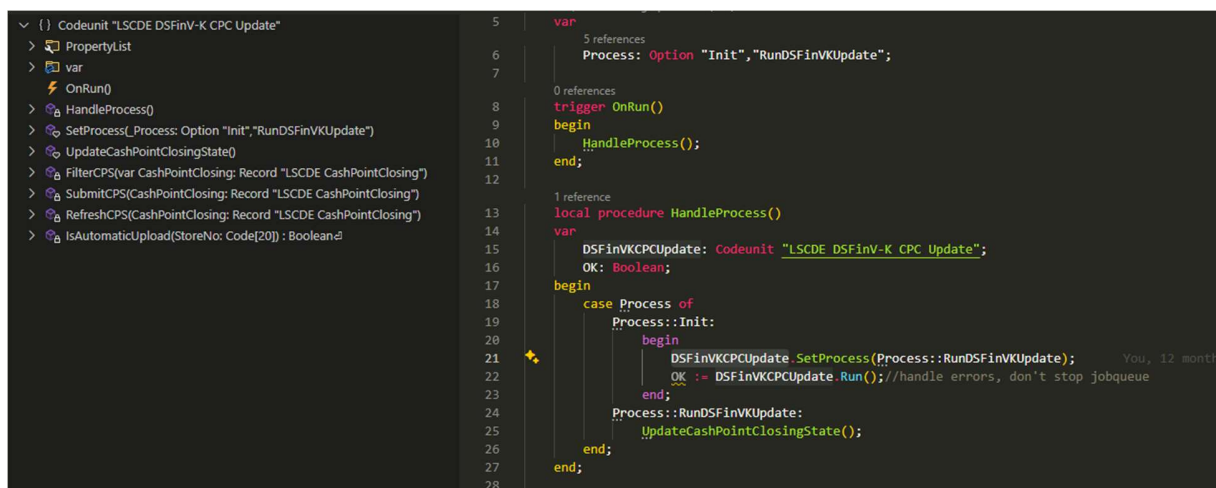


ábra 26: Service Locator

5.10. Feladatvárólista

Létrehoztam a DSFinV-K CPC Update codeunit-ot, melynek a célja, hogy ütemezett feladatként be tudjuk állítani és frissíteni a beküldött Cash Point Closing rekordok státuszát. Fontos, hogy kezeljük a hibákat, mivel ha hiba státuszra fut a feladatvárólista, akkor az ütemezés leáll és manuálisan kell újraindítani.

Hogy biztosítsam a hibamentes futást így létrehoztam egy globális változót, process néven. Ez az option változó két állapottal rendelkezik, Init és RunDSFinVKUpdate. Alapértelmezetten Init státuszban áll, amikor a feladatvárólista futtatja az OnRun triggert, akkor egy másik DSFinV-K CPC Update példányon beállítom, hogy a process legyen RunDSFinVKUpdate és kondicionálisan futtatom azt a példányt is, mint ahogy a 27. ábrán látható.



ábra 27: DSFinV-K CPC Update[8]

5.11. DSFinV-K export

Az export funkcionalitást a fiskaly dashboard-on keresztül érhetjük el, jelenleg ezt nem implementálom.

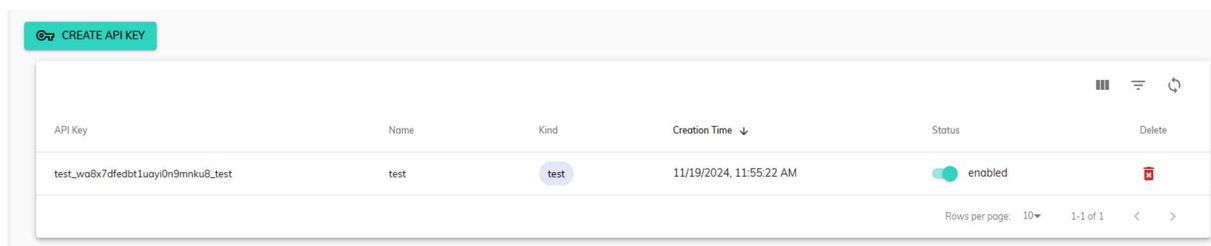
6. TESZTELÉS

Először manuálisan tesztelem a funkcionalitást, majd unit teszteket készítek.

6.1. Manuális

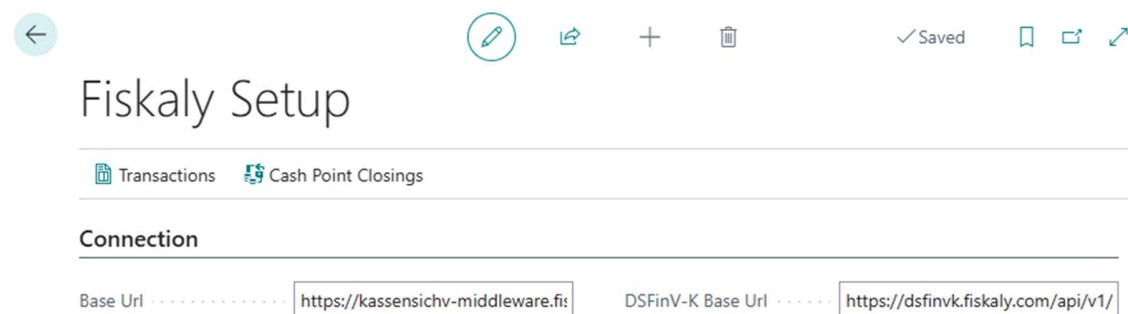
A manuális teszteket SaaS és lokális környezeten is el tudom végezni, én egy SaaS környezetet választottam, mivel ez már rendelkezésemre állt a fejlesztés alatt. Készítettem egy új Company-t melybe frissen importáltam a demo adatokat. Az alapvető konfigurációk után végrehajtottam a DSFinV-K specifikus beállításokat.

Mivel már rendelkezem fiskaly fiókkal, így a regisztrációt nem hajtottam végre újból, viszont generáltam egy új Key-t és Secret-et ahogy a 28. ábrán látszódik. A Secret-et egy Notepad-be elmentve tárolom, mivel csak a létrehozásnál látszódik.



ábra 28: fiskaly API key teszteléshez[8]

Megnyitom a Fiskaly Setup page-et, hogy beillessze a setup rekordot ahogy az 29. ábrán átható.



ábra 29: Fiskaly Setup[8]

Az url mezők automatikusan töltődnek a rekord beszúrásakor.

Következő lépésként beállítom a Store Card-on a Fiskaly API Key-t és Secret-et, majd engedélyezem a szolgáltatásokat. Az Upload Type-ot manual-ra állítom. Az új TSS generálása és inicializálása után a 30. ábrán látható az eredmény.

Fiskaly Setup

Fiskaly API Key: test_wa8x7dfedbt1uay0n9mku8_test

Fiskaly API Secret: *****

Enabled: ☒

Upload Type: Manual

Use DSFinV-K: ☒

Fiskaly TSS Setup | [Generate New TSS](#) | [Initial TSS](#) | [Refresh TSS](#) | [Disable TSS](#)

TSS ID	State	Error Text	Created At
→ [0ee87dca-cb81-49be-8f01-a95929d57866]	INITIALIZED		11/19/2024 1:13 PM

ábra 30: Store karton beállítások[8]

Meggyőződtem, hogy a fiskaly Dashboard-on is megjelent a TSS mint a 31. ábrán látható.

Ferenc Szilágyi
ferencszilagyi7@gmail.com
Dev1

ENABLE LIVE

fiskaly Solutions

- Germany
 - Dashboard
 - SIGN DE V2 (Germany)
 - Technical Security...**
 - Clients
 - Transactions
 - Exports
 - SIGN DE V1 (deprecated)
 - DSFinV-K

CREATE NEW TSS

ID	Version	Description	State	Transaction Counter	Signature Counter	Certificate Serial No.
> [0ee87dca-cb81-49be-8f01-a95929d57866]	v2.1.19	fiskaly sign cloud-TSE (0ee87dca-cb81-49be-8f01-a95929d57866)	INITIALIZED		37	8ee063fbddc3dc3a613030454217bc6f24d96d9e2f3188050020789763103H
> [38a50098-191f-47be-8a05-214cd19cf939]	v2.1.18	fiskaly sign cloud-TSE (38a50098-191f-47be-8a05-214cd19cf939)	DELETED	5	80	da41d9778099c055f3d2083a3768310b7ca19963feb3250b939bbf898b4a21
> [1558459c-fc9b-4cd7-8780-05e44562da71]	v2.1.18	fiskaly sign cloud-TSE (1558459c-fc9b-4cd7-8780-05e44562da71)	DELETED	1	72	ca35d36b2d24655af3d2ab035a967190d9d27239f3d7712e78f2679223f80f6
> [c8411460-...]		fiskaly sign cloud-TSE				

ábra 31: fiskaly Dashboard TSS[8]

A Store beállítások után átnavigálok a kapcsolódó POS Terminálokra, és elkészítem a Cash Register-t. Első sorban klienst kell generálnom, ez még a Sign DE V2 részét képezi. Generálás után látható a 32. és 33. ábrán az elkészített kliens, Business Central és fiskaly oldalon is.

Sign DE V2 | [Generate Client ID](#) | [Deregister Client](#)

POS Terminal No. ↑	Store ID	TSS ID	Client ID	Serial No.	Client State	Tss State	Error Text
→ [E0001]	S0001	[0ee87dca-cb81-49be-8f01-a95929d57866]	[0782fe21-b73d-4a42-98a2-c51b6a0962c3]	ERS 03cd5d20-7bb0-4031-a337-93...	REGISTERED	INITIALIZED	

ábra 32: fiskaly kliens BC[8]

ID	Version	Serial No.	State	Creation Time	Update Time	TSS ID
0782fe21-b73d-4a42-98a2-c51b6a8962c3	v2.1.18	ERS 03cd5420-7bb0-4031-a337-938db616f8a	REGISTERED	11/19/2024, 1:20:11 PM		0ee87dca-cb81-49be-8f01-a95929d57866
5e78b688-633a-4a25-b6b4-347cd8a30adb	v2.1.18	ERS c2343b54-6042-46b9-947f-62f1e99d602	REGISTERED	8/28/2024, 4:05:53 AM		36e50098-191f-47be-8ef0-214cd19cf939
3b0c2d83-a4bc-4546-84d7-a0ca2168f638	v2.1.18	ERS f6920a20-68be-4886-bf26-73757280eb42	REGISTERED	8/26/2024, 5:58:37 AM		1568459c-4ef8-4cd7-8780-05e44562db71
ec389733-394c-4890-adc7-88b8089a0ac4	v2.1.18	ERS a050c2b7-92e1-498f-9707-637d07f6afa	REGISTERED	8/21/2024, 7:02:26 PM		cb41f460-2534-43ab-9424-453bad7870e4
4af19703-21af-4644-ba9c-f120da27e3ab	v2.1.18	ERS ea2a2de1-cfed-43c5-9bd2-c5ee6b2a556	REGISTERED	8/21/2024, 6:30:40 PM		cf6a11c2-1a31-4a9e-bf88-a8ac1f69e7f6
f7878288-40eb-40d9-8038-f2e6a076b0d5	v2.1.18	ERS a7776a74-6093-44f2-ba41-96cc5225cb2c	REGISTERED	8/21/2024, 6:29:50 PM		e5303684-3640-40d1-9dfb-88073344298d

ábra 33: fiskaly kliens Dashboard[8]

Az Insert Cash Register gombra kattintva felugrik a 34. ábrán látható Cash Register oldal, amivel inicializálhatom az új pénztárgépet.

Cash Register Card - P0001 · 0

Manage | Show as menu

General

Type: MASTER | Model: Test Model

Client ID: {0782fe21-b73d-4a42-98a2-c51b6a8962c3} | Software Brand: SW Brand

TSS ID: {0ee87dca-cb81-49be-8f01-a95929d57866} | Software Version: 1.0

Brand: Test Brand | Base Currency Code: EUR

OK Cancel

ábra 34: Cash Register beszúrása, frissítése[8]

Az OK gombra kattintva amennyiben sikeres a művelet, beszúrjuk a rekordot és létrehozuk a Cash Registert Business Central (35. ábra) és fiskaly oldalon (36. ábra) is.

Type	Status	Client ID	TSS ID	Brand	Model	Software Brand	Software Version	Base Currency Code
→ MASTER	Active	{0782fe21-b73d-4a42-98a2-c51b6a8962c3}	{0ee87dca-cb81-49be-8f01-a95929d57866}	Test Brand	Test Model	SW Brand	1.0	EUR

ábra 35: Cash Register létrehozva BC[8]

ID	Version	Cash Register Type	Master Client ID	Revision	TSS ID	Creation Time	Update Time
0782fe21-b73d-4a42-98a2-c51b6a8962c3	v1.26.0	MASTER		0	0ee87dca-cb81-49be-8f01-a95929d57866	11/19/2024, 1:25:03 PM	11/19/2024, 1:25:03 PM
5e78f688-533a-4a25-b6b4-347cd8e30ada	v1.25.2	MASTER		0	36e50098-191f-47be-8ef0-214cd19cf939	8/28/2024, 4:07:13 AM	8/28/2024, 4:07:13 AM
3b9b2d83-e4bc-4546-84d7-a0ca2168638	v1.25.2	MASTER		0	1568459c-fe9b-4cd7-8780-05e44562da71	8/26/2024, 6:02:19 AM	8/26/2024, 6:02:19 AM

ábra 36: Cash Register létrehozva fiskaly[8]

Teszt képpen készítettem egy slave pénztárat is egy másik POS Terminálhoz. Látszódik a 37. ábrán, hogy a típusa SLAVE_WITH_TSS.

02c417a0-df23-40c2-b548-116f896cb1a3	v1.26.0	SLAVE_WITH_TSS	0782fe21-b73d-4a42-98a2-c51b6a8962c3	0	0ee87dca-cb81-49be-8f01-a95929d57866	11/19/2024, 1:27:51 PM	11/19/2024, 1:27:51 PM
--------------------------------------	---------	----------------	--------------------------------------	---	--------------------------------------	------------------------	------------------------

ábra 37: Slave Cash Register[8]

POS VAT Code beállításoknál a DSFinV-K Export ID-t állítom be custom range-re ahogy a 38. ábra mutatja.

VAT Code T	Description	VAT %	Fiskaly VAT Rate	DSFinV-K Export ID	Fiscal ID
A	No VAT	0.00	NULL	1000	0
B	Reduced	10.00	REDUCED_1	1003	0
C	Standard	25.00	NORMAL	1002	0

ábra 38: POS VAT Code beállítás[8]

Ez a módosítások után az oldalon található Generate VAT Definitions action-nel szinkronizálom a rekordokat.

POS Tranzakció létrehozása után a Cash Register Client ID be van állítva a Fiskaly Sales Transaction rekordon ahogy a 39. ábrán is látható.

Store No.	POS Terminal No.	Transaction No.	Fiskaly State	Receipt No.	Fiskaly Transaction Number	Transaction ID	Revision	Document Amount	Cash Amount	Non-Cash Amount	Status	Error Te
S0001	P0001	2	FINISHED	00000P000...	1	(29a75301-d267-4753-88f0-4b...	2	30.00	30.00	0.00	Completed	

ábra 39: Fiskaly Sales Transaction[8]

A sikeres Statement Posting után létrejön a Cash Point Closing, melyet a submit gombbal be tudunk küldeni, majd a refresh gomb használatával tudjuk frissíteni az állapotát, mint ahogy a 40., 41. és 42. ábrán látható.

Closing ID	Cash Point Closing Export ID	State	Error	Client ID	Business Date	First Transaction Exp. ID	Last Transaction Exp. ID	Export Creation Date	Time Creation	Time Update
a808ff94-93ec-4a6f-9aca-ad0...		PENDING		[0782fe21-b73d-4a42-98a2-c5...		1	1	0	1732020960	0

ábra 40: Pending Cash Point Closing[8]

Closing ID	Cash Point Closing Export ID	State	Error	Client ID	Business Date	First Transaction Exp. ID	Last Transaction Exp. ID	Export Creation Date	Time Creation	Time Update
a808ff94-93ec-4a6f-9aca-ad0...		COMPLETED		[0782fe21-b73d-4a42-98a2-c5...	11/19/2024	1	1	1732020967	1732020960	1732020960

ábra 41: Completed Cash Point Closing[8]

Closing ID	Version	Status	Error	Client ID	Number	Creation	Business Date	Creation Time
u808ff94-93ec-4a6f-9aca-ad05dfc43391	1.26.0	COMPLETED		[0782fe21-b73d-4a42-98a2-c51b6a8962c3	1	11/19/2024, 1:56:07 PM	2024-11-19	11/19/2024, 1:56:00 PM
7478deb4-				5e78fd88-				

ábra 42: Completed Cash Point Closing fiskaly[8]

A feladatvárólistát beállítottam a 43. ábrán látható módon és 5 percenként futtatom.

Codeunit · 10044595 · LSCDE DSFinV-K CPC Update

Home Job Queue

Set Status to Ready Set On Hold Restart Run once (foreground) Show Error

General Show less

Object Type to Run	Codeunit	Maximum No. of Atte...	0
Object ID to Run	10044595	Rerun Delay (sec.)	0
Object Caption to Run	LSCDE DSFinV-K CPC Update	Last Ready State	11/19/2024 1:59 PM
Description	LSCDE DSFinV-K CPC Update	Earliest Start Date/Ti...	
Parameter String		Expiration Date/Time	
Job Queue Category ...		Job Timeout	12 hours
Priority	Normal	Status	On Hold
User ID	LSRETAIL\FERENC.SZILAGYI		

ábra 43: Feladatvárólista beállítás[8]

A teszteléséhez létrehoztam egy újabb tranzakciót és Statement Posting-ot, majd a Store-on átállítottam automatic-ra az Upload Type mezőt.

A rendszer tesztelése során megállapítottam, hogy az elvárt módon működik.

6.2. Tesztautomatizálás

PowerShell script segítségével létrehozom a Business Central példányt, amivel dolgozni fogok. A 44. ábrán látható paramétereket szükséges beállítani:

```
1 $ErrorActionPreference = 'stop'
2
3 Import-Module LsPackageTools\Workspace
4
5 # --- Parameters -----
6 $branchName = 'release/release'
7 $countryCode = 'DE'.ToLower()
8 $instanceName = "DSFinVK_dev"
9
10 $useBcDemoDatabase = $true
11 # If connection string is set, will install on existing database.
12 $connectionString = ''
13 $connectionString = $null
14
15 # -----
16
```

ábra 44: Helyi környezet létrehozása script 1[8]

A Server Instance neve DSFinVK_dev és az országcód pedig DE, mivel a német gyári lokalizációra van szükségünk.

A script paraméterezés után futtatható és az alábbi sor létre fogja hozni a kívánt példányt.

```
82
83 # $packages | Get-UscUpdates
84 $packages | Install-UscPackage -InstanceName $instanceName -Arguments $Arguments
```

ábra 45: Helyi környezet létrehozása script 2[8]

A tesztelés megkezdése előtt publikálom a gyári teszt könyvtárakat szintén PowerShell script segítségével ahogy a 46. ábrán látszik. Ezek a bővítmények a hivatalos DVD.zip fájl mélyén találhatóak meg.

```
1 #TEST DEPENDENCIES + #####
2 $ServerInstance = 'DSFinVK_dev'
3 $SourcePath = 'C:\DEVELOPMENT\AL\Source\Dynamics.365.BC.18056.W1.DVD\Applications\testframework\TestLibraries\contains all apps and .dll-s
4 $MockDllSrcPath = 'C:\DEVELOPMENT\AL\Source\Dynamics.365.BC.18056.W1.DVD\Test Assemblies\Mock Assemblies'
5 $TestAppPath = 'C:\DEVELOPMENT\AL\Source\Dynamics.365.BC.18056.W1.DVD\Applications\BaseApp\Test'
6
7 Import-Module ('C:\Program Files\LS Retail\Update Service\Instances\' + $ServerInstance + '\Service\Microsoft.Dynamics.Nav.Management.psml')
8 Import-Module ('C:\Program Files\LS Retail\Update Service\Instances\' + $ServerInstance + '\Service\Microsoft.Dynamics.Nav.Apps.Management.psdl');
9 Import-Module ('C:\Program Files\LS Retail\Update Service\Instances\' + $ServerInstance + '\Service\NavAdminTool.ps1');
10
11 #Get-NAVServerInstance
12 #Get-NAVApplication -ServerInstance $ServerInstance
13 #Get-NAVAppInfo -ServerInstance $ServerInstance
14
15 Copy-Item -Path ($MockDllSrcPath+'MockTest.dll') -Destination ('C:\Program Files\LS Retail\Update Service\Instances\' + $ServerInstance + '\Service\Add-ins\MockTest.dll')
16
17 #DO NOT SKIP RESTART
18 Restart-NAVServerInstance -ServerInstance $ServerInstance
19
20 #PUBLISHING APPS IN ORDER
21
22 Publish-NAVApp -ServerInstance $ServerInstance -Path ($SourcePath + '\Assert\Microsoft.Library Assert.app') -Scope Global
23 Publish-NAVApp -ServerInstance $ServerInstance -Path ($SourcePath + '\Any\Microsoft.Any.app') -Scope Global
24 Publish-NAVApp -ServerInstance $ServerInstance -Path ($SourcePath + '\Variable Storage\Microsoft.Library Variable Storage.app') -Scope Global
25 Publish-NAVApp -ServerInstance $ServerInstance -Path ('C:\DEVELOPMENT\AL\Source\Dynamics.365.BC.18056.W1.DVD\Applications\system application\test\Microsoft_System Application Test Library.app') -Scope Global
26 Publish-NAVApp -ServerInstance $ServerInstance -Path ('C:\DEVELOPMENT\AL\Source\Dynamics.365.BC.18056.W1.DVD\Applications\system application\test\Microsoft_System Application Test.app') -Scope Global
27 Publish-NAVApp -ServerInstance $ServerInstance -Path ('C:\DEVELOPMENT\AL\Source\Dynamics.365.BC.18056.W1.DVD\Applications\BaseApp\Test\Microsoft_Tests-TestLibraries.app') -Scope Global -SkipVerification -Force
28
29
```

ábra 46: Teszt library installálás 1[8]

Publikálás után lefuttatom a Sync és Install parancsot a 47. ábra szerint.

```

29
30 #INSTALL
31 ##Library Assert
32 Sync-NAVApp -ServerInstance $ServerInstance -AppId "dd0be2ea-f733-4d65-bb34-a28f4624fb14"
33 Install-NAVApp -ServerInstance $ServerInstance -AppId "dd0be2ea-f733-4d65-bb34-a28f4624fb14"
34 ##Any
35 Sync-NAVApp -ServerInstance $ServerInstance -AppId "e7320ebb-08b3-4406-b1ec-b4927d3e280b"
36 Install-NAVApp -ServerInstance $ServerInstance -AppId "e7320ebb-08b3-4406-b1ec-b4927d3e280b"
37 ##Variable Storage
38 Sync-NAVApp -ServerInstance $ServerInstance -AppId "5095f467-0a01-4b99-99d1-9ff1237d286f"
39 Install-NAVApp -ServerInstance $ServerInstance -AppId "5095f467-0a01-4b99-99d1-9ff1237d286f"
40 ##MSA Test Library
41 Sync-NAVApp -ServerInstance $ServerInstance -AppId "9856ae4f-d1a7-46ef-89bb-6ef056398228"
42 Install-NAVApp -ServerInstance $ServerInstance -AppId "9856ae4f-d1a7-46ef-89bb-6ef056398228"
43 ##MSA Test
44 Sync-NAVApp -ServerInstance $ServerInstance -AppId "0d60b215-6ee1-4789-8e53-866cfa50c23c"
45 Install-NAVApp -ServerInstance $ServerInstance -AppId "0d60b215-6ee1-4789-8e53-866cfa50c23c"
46 ##Tests-TestLibraries
47 Sync-NAVApp -ServerInstance $ServerInstance -AppId "5d86850b-0d76-4eca-bd7b-951ad998e997"
48 Install-NAVApp -ServerInstance $ServerInstance -AppId "5d86850b-0d76-4eca-bd7b-951ad998e997"
49

```

ábra 47: Teszt library installálás 2[8]

Jelenleg még a német lokalizációhoz nincs rendes teszt alkalmazásunk, csak egy helyőrző, így inicializálok egy projektet az AL:Go! VS Code funkció segítségével és beállítom a következő függőségeket.

```

"dependencies": [
  {
    "id": "LS Central",
    "publisher": "LS Retail",
    "name": "LS Central",
    "version": "24.0.0.0"
  },
  {
    "id": "Local Functionality for LS Central (Germany)",
    "publisher": "LS Retail",
    "name": "Local Functionality for LS Central (Germany)",
    "version": "24.0.0.0"
  },
  {
    "id": "LS Central Autotests",
    "publisher": "LS Retail",
    "name": "LS Central Autotests",
    "version": "24.0.0.0"
  },
  {
    "id": "Library Assert",
    "publisher": "Microsoft",
    "name": "Library Assert",
    "version": "24.0.0.0"
  },
  {
    "id": "Tests-TestLibraries",
    "publisher": "Microsoft",
    "name": "Tests-TestLibraries",
    "version": "24.0.0.0"
  }
]

```

ábra 48: Teszt alkalmazás függőségek[8]

VS Code segítségével publikálom a teszt alkalmazást és ellenőrzöm, hogy a közzététel sikeres-e. Amint ezt sikeresen végrehajtottam, megkezdem a tesztesetek fejlesztését.

6.2.1. Autentikáció

Az Authentication-UT codeunit egy sor unit tesztet tartalmaz, amelyek célja az Authentication codeunit működésének ellenőrzése. Az első teszt, az Authorize_Invokes_GetToken, azt vizsgálja, hogy az Authorize metódus megfelelően hívja-e meg a TokenService Mock codeunit GetToken

metódusát, inicializálja a szükséges beállításokat, meghívja az Authorize metódust, majd ellenőrzi, hogy a GetToken valóban meghívásra került-e.

A GetToken_Store_No_Undefined_ThrowsError teszt azt biztosítja, hogy a GetToken metódus hibát dob, ha a Store rekord "No." mezője nincs definiálva. Az asserterror kulcsszóval várható hibát definiál, majd ellenőrzi, hogy a hibaüzenet tartalmazza a "No." mező megnevezését.

A GetToken_Store_APIKey_Undefined_ThrowsError teszt azt vizsgálja, hogy a GetToken metódus hibát dob, ha a Store rekord "API Key" mezője nincs beállítva; a Store rekord "No." mezőjét beállítja, definiálja a várható hibát és ellenőrzi, hogy a hibaüzenet tartalmazza az "API Key" mező megnevezését.

A GetToken_Invokes_PostGetTokenRequest teszt ellenőrzi, hogy a GetToken metódus megfelelően hívja-e meg a Common Utilities Mock codeunit GetPassword metódusát, valamint a TokenService Mock codeunit PostGetTokenRequest metódusát; ehhez inicializálja a szükséges beállításokat, beállítja a Store rekord mezőit, majd ellenőrzi, hogy mindkét metódus a megfelelő paraméterekkel lett-e meghívva.

A PostGetTokenRequest teszt azt vizsgálja, hogy a PostGetTokenRequest metódus a megfelelő URL-re küld-e POST kérést a helyes tartalommal, és hogy visszaadja-e a várt JWT token.

A teszt inicializálja a szükséges beállításokat, beállítja az alap URL-t a DSFinVKMgt Mock codeunitban, és a Mock HttpClient codeunit segítségével szimulálja a HTTP választ, majd ellenőrzi, hogy a POST kérés a megfelelő URL-lel és tartalommal került-e elküldésre, valamint hogy a visszakapott JWT token megfelel-e az elvárásoknak.

6.2.2. Statement Posting

Az alábbi tesztek célja a CreateCashPointClosing és a FinalizeCashPointClosing metódusok különböző feltételek melletti működésének ellenőrzése.

A CreateCashPointClosing_DSFinVKDisabled_Exits teszt azt vizsgálja, hogy a CreateCashPointClosing metódus idő előtt kilép, és nem hívja meg a CheckCashPointClosing metódust, ha a DSFinV-K le van tiltva; ezt a szükséges mock-ok beállításával ellenőrzi.

A CreateCashPointClosing_NoMasterForStore_Exits teszt azt biztosítja, hogy a CreateCashPointClosing metódus akkor is idő előtt kilép, ha nincs fő pénztárgép a bolt számára, és ellenőrzi, hogy a CheckCashPointClosing nem került meghívásra.

A CreateCashPointClosing_Completes teszt ellenőrzi, hogy a CreateCashPointClosing metódus sikeresen lefut, ha a DSFinV-K engedélyezve van és van fő pénztárgép; beállítja a szükséges mock-okat és adatfolyamot, majd megerősíti, hogy a CheckCashPointClosing és az InsertFromStatement a megfelelő paraméterekkel került meghívásra.

A `FinalizeCashPointClosing_DSFinVKDisabled_Exits` teszt azt vizsgálja, hogy a `FinalizeCashPointClosing` metódus idő előtt kilép, és nem hívja meg a `Finalize` metódust, ha a `DSFinV-K` le van tiltva; ezt a szükséges mock-ok beállításával ellenőrzi.

A `FinalizeCashPointClosing_GetLastCashPointClosing_False_Exits` teszt azt biztosítja, hogy a `FinalizeCashPointClosing` metódus akkor is kilép, ha a `GetLastCashPointClosing` hamis értéket ad vissza, és ellenőrzi, hogy a `Finalize` nem került meghívásra.

A `FinalizeCashPointClosing_FiskalyAutomaticUpload_False_Exits` teszt azt vizsgálja, hogy a `FinalizeCashPointClosing` metódus idő előtt kilép-e, ha a `FiskalyAutomaticUpload` hamis értéket ad vissza, és ellenőrzi, hogy a `Finalize` nem került meghívásra.

A `FinalizeCashPointClosing_Completes` teszt ellenőrzi, hogy a `FinalizeCashPointClosing` metódus sikeresen lefut-e, ha a `DSFinV-K` engedélyezve van, a `GetLastCashPointClosing` igaz értéket ad vissza, és a `FiskalyAutomaticUpload` is igaz értéket ad vissza; ellenőrzi továbbá, hogy a `Finalize` a megfelelő `Posting No.` mezővel került meghívásra.

7. ÉRTÉKELÉS

Az LS Central DE rendszer DSFinV-K integrációjának fejlesztése sikeresnek bizonyult, mivel az elkészült funkcionális minden teszt során megfelelt a projekt előzetes célkitűzéseinek és a német pénzügyi szabályozási követelményeknek. A kulcselemek – mint a Cash Register regisztráció, tranzakciók beküldése és pénztárzárások – az első teszteken és a felhasználói ellenőrzéseken is átmentek, amelyek megerősítették a DSFinV-K szabványokkal való kompatibilitást. A fejlesztett felhasználói felület és adatbázis bővítések hozzájárultak a hatékony és gördülékeny munkafolyamat kialakításához, ezáltal biztosítva a szabályozási követelmények egyszerű és problémamentes teljesítését a felhasználók számára. A megoldás éles környezetben történő telepítése további bizalmat adott a rendszer megbízhatóságához és stabil működéséhez.

A fiskális DSFinV-K szolgáltatás integrációja az LS Central DE rendszerbe egy átfogó megoldást kínál a német tranzakciós jelentési szabályozások betartásához. A Microsoft Dynamics 365 Business Central alapfunkcionálisának kihasználásával sikerült egy biztonságos és hatékony rendszert létrehozni, amely biztosítja az adatintegritást, a jogszabályoknak való megfelelést és lehetővé teszi a pénzügyi tranzakciók szisztematikus kezelését. Az integráció moduláris kialakítása lehetőséget nyújt a jövőbeli bővítésekhez és a változó szabályozási környezethez való alkalmazkodáshoz, ezzel is stabil alapot biztosítva a vállalkozások számára pénzügyi műveleteik hosszú távú kezeléséhez.

A rendszertervezés során külön figyelmet fordítottunk az automatizálásra és a biztonságos adatkezelésre. Ez nemcsak leegyszerűsíti a tranzakciós jelentési folyamatokat, hanem hozzájárul a pénzügyi rendszerek átláthatóságához és megbízhatóságához is, ami elengedhetetlen a szabályozásoknak való megfeleléshez. Ahogy a vállalkozások az egyre bonyolultabb szabályozási elvárásoknak próbálnak megfelelni, a DSFinV-K szolgáltatáshoz hasonló innovatív megoldások kulcsszerepet játszanak a megfelelés elősegítésében és az operatív hatékonyság növelésében a kiskereskedelmi szektorban.

8. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az LS Central DE rendszer DSFinV-K integrációjának további fejlesztése során kiemelt figyelmet kaphat az exportálási funkcionalitás közvetlen beépítése a rendszerbe. Jelenleg a fiskaly Dashboard-on keresztül érhető el az export, azonban a jövőben, ügyféligények függvényében célszerű lenne ezt a funkciót közvetlenül integrálni az LS Central felületébe. Ez nemcsak a folyamat gyorsítását és egyszerűsítését segítené elő, hanem a felhasználói élményt is javítaná, mivel a felhasználóknak nem kellene külső felületre navigálniuk az export elvégzéséhez.

A rendszer moduláris felépítésének köszönhetően az integrációs kód és a szabályozási logika könnyen módosítható, ha a jogszabályok vagy az ügyfélkövetelmények megváltoznak. Mivel ez a modul egy jogi előírásnak és ügyféligénynek felel meg, korlátozott lehetőség áll rendelkezésre az egyedi fejlesztési ötletek megvalósítására. Azonban amennyiben az előírások vagy a felhasználói igények változnak, kötelességünk ezeket az új követelményeket a modulba integrálni, biztosítva ezzel a folyamatos jogi megfelelést és ügyfélelégedettséget.

A továbbfejlesztés másik fontos területe a tesztelési lefedettség (code coverage) növelése. Az egység és integrációs tesztek számának bővítése lehetővé tenné, hogy a rendszer minden fontos funkcionalitása megbízhatóan ellenőrzött legyen, csökkentve ezzel a hibák lehetőségét, valamint növelve a rendszer stabilitását és minőségét. A code coverage javítása különösen fontos ebben a rendszerben, mivel a szabályozási megfelelés és az adatbiztonság kiemelt fontossággal bír. A lefedettség növelésével biztosítható, hogy a jövőbeli frissítések és módosítások zavartalanul illeszkedjenek a meglévő kódbázisba.

A jelenlegi verziót partnereink már tesztelik és bevezetésre került ügyfeleiknél. A bevezetés során összegyűjtött visszajelzések – legyenek azok pozitívak vagy negatívak – értékes információkat nyújtanak a rendszer további finomításához és fejlesztéséhez. Az ügyfélvisszajelzések alapján lehetőségünk nyílik arra, hogy a rendszer működését és hatékonyságát az ügyféligények és a gyakorlati tapasztalatok alapján tovább optimalizáljuk.

9. ÖSSZEGZÉS

A fejlesztés során alaposan megismerkedtem a fiskaly DSFinV-K szolgáltatás integrálásának lépéseivel, a munka során új módszereket és nyelvi elemeket alkalmaztam. Kiemelt tapasztalatot szereztem az interfészek implementálásában, amely az AL nyelv viszonylag új funkciója és eddigi munkámban nem használtuk. Ennek részleteiben egy Microsoft MVP nyújtott szakmai támogatást, mivel a dokumentáció korlátozott, ezért a legjobb megoldásokhoz tapasztalati alapú útmutatásra volt szükségem.

A fejlesztési folyamatot a Scrum módszertan irányelvei alapján szerveztük, ami megfelelő rugalmasságot biztosított számomra. Ez lehetőséget adott arra, hogy konzultáljak a fiskaly támogatói csapattal és a belső csapattal is, elősegítve a minőségi megoldás kialakítását. A fejlesztés release előtti tesztelése során kapott visszajelzések alapján módosítottam a felhasználói felület bizonyos elemeit, hogy a rendszer kezelése még felhasználóbarátabb legyen. Fontos tapasztalat volt számomra, hogy a nem fejlesztő kollégák más szemszögből vizsgálják az alkalmazást, és javaslataik jelentősen hozzájárulnak a könnyebb használhatósághoz.

Ez a projekt számomra azért is különleges, mert most először kezeltem egy teljes EPIC fejlesztési ciklust az elejétől a végéig. Az analízis folyamata is újdonság volt, korábban jellemzően kész fejlesztési tervet kaptam. A fiskaly csapatával való közvetlen együttműködés segítette a pontosabb igényfelmérést és megvalósítást.

A projekt során a cég új szabályzata miatt kiemelt figyelmet kellett fordítanom az automatizált tesztelésre, beleértve az integrációs és unit tesztek írását is, amelyek a rendszer működését alátámasztják. Ez változtatásokat igényelt az eddigi fejlesztési módszereinkben, beleértve az interfészek implementálását is. Új munkatársként különösen értékelem a csapat bizalmát, hogy rám bízta ezt a komplex fejlesztést, amely során lehetőségem nyílt hozzájárulni az új tesztelési és fejlesztési módszerek bevezetéséhez.

A fejlesztés sikeresen lezárult, és a megoldás mostantól elérhető az LS Central német lokalizációjában, támogatva a helyi jogszabályoknak való megfelelést és a felhasználók hatékony munkafolyamatait.

10. SUMMARY

During the development, I learned the steps of the integration of the fiskaly DSFinV-K service, and used new methods and language elements. I have gained considerable experience in implementing interfaces, a relatively new feature of the AL language and one that has not been used in my previous work. A Microsoft MVP provided technical support in the details of this, as documentation is limited, I needed experience-based guidance for the best solutions.

The development process was organised according to the guidelines of the Scrum methodology, which gave me sufficient flexibility. This gave me the opportunity to consult with the fiskaly support team as well as the internal team, helping to develop a quality solution. Based on the feedback received during the pre-release testing of the development, I modified certain elements of the user interface to make the system more user-friendly to use. It was an important experience for me to see non-developer colleagues looking at the application from a different perspective, and their suggestions contributed significantly to the ease of use.

This project was also special for me because it was the first time I managed a complete EPIC development cycle from start to finish. The process of analysis was also new, in the past I typically received a ready-made development plan. Working directly with the fiskaly team helped to ensure a more accurate needs assessment and implementation.

During the project, the new company policy required me to pay special attention to automated testing, including writing integration and unit tests to validate the system's functionality. This required changes to our existing development methods, including the implementation of interfaces. As a new employee, I particularly appreciate the team's confidence in me to take on this complex development project, which gave me the opportunity to contribute to the introduction of new testing and development methods.

The development has been successfully completed and the solution is now available in the German localization of LS Central, supporting local compliance and efficient workflows for users.

IRODALOMJEGYZÉK

- [1] „About LS Retail | LS Retail software solutions”. Elérés: 2023. november 25. [Online]. Elérhető: <https://www.l retail.com/about-us>
- [2] „SIGN DE API V2 | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/kassensichv/v2>
- [3] „What Is a POS System and How Does It Work?” Elérés: 2023. december 2. [Online]. Elérhető: <https://www.investopedia.com/terms/p/point-of-sale.asp>
- [4] „POS Systems | Point of Sale for Small Businesses | Square”. Elérés: 2023. december 2. [Online]. Elérhető: <https://squareup.com/us/en/point-of-sale>
- [5] „What Is Shopify and How Does It Work? (2024)”. Elérés: 2023. december 2. [Online]. Elérhető: <https://www.shopify.com/blog/what-is-shopify>
- [6] „Leading Point of Sale (POS) & Commerce Platform | Lightspeed”. Elérés: 2023. december 2. [Online]. Elérhető: <https://www.lightspeedhq.com/home/>
- [7] „LS Central | Retail POS Software | All-in-one retail management system”. Elérés: 2023. november 25. [Online]. Elérhető: <https://www.l retail.com/products/ls-central-for-retail>
- [8] Szilágyi Ferenc, „Saját készítésű ábra”.
- [9] „Welcome to Microsoft Dynamics 365 Business Central - Business Central | Microsoft Learn”. Elérés: 2023. november 25. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/welcome>
- [10] „Germany”. Elérés: 2024. november 6. [Online]. Elérhető: https://help.lscentral.l retail.com/Content/Implementation-Guide/Localization/Germany.htm?tocpath=Local%20Functionality%7CGermany%7C___0
- [11] „Fiscalization solutions made for people”. Elérés: 2024. november 6. [Online]. Elérhető: <https://www.fiskaly.com/>
- [12] „Information platform for the Cash Register Security Ordinance”. Elérés: 2024. november 18. [Online]. Elérhető: <https://kassensichv.net/en>
- [13] „DSFinV-K: The Standard for Cash Register Inspections and Data Export”. Elérés: 2024. november 18. [Online]. Elérhető: <https://kassensichv.net/en/dsfinv-k>
- [14] „What is the RKS?” Elérés: 2024. november 18. [Online]. Elérhető: <https://www.fiskaly.com/blog/what-is-the-rksv>
- [15] „TicketBAI API • Integrate TBAI for Spain in your software”. Elérés: 2024. november 18. [Online]. Elérhető: <https://www.fiskaly.com/signes/ticketbai>

- [16] „[FREE] fiskalcheck App by fiskaly: scan & validate receipts”. Elérés: 2024. november 18. [Online]. Elérhető: <https://www.fiskaly.com/fiskalcheck-app>
- [17] „One digital receipt API, many countries”. Elérés: 2024. november 18. [Online]. Elérhető: <https://www.fiskaly.com/digital-receipt>
- [18] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#section/Introduction>
- [19] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 5. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Authentication>
- [20] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/VAT-Definition>
- [21] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Purchaser-Agency>
- [22] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Cash-Register>
- [23] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Cash-Point-Closing>
- [24] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 6. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Exports>
- [25] „Source Code Management | W3Docs Git Online Tutorial”. Elérés: 2023. november 27. [Online]. Elérhető: <https://www.w3docs.com/learn-git/source-code-management.html>
- [26] „Git”. Elérés: 2023. november 27. [Online]. Elérhető: <https://git-scm.com/>
- [27] „10 Git Commands Every Developer Should Know”. Elérés: 2023. november 28. [Online]. Elérhető: <https://www.freecodecamp.org/news/10-important-git-commands-that-every-developer-should-know/>
- [28] „Version Control Software: An Overview | Bitbucket”. Elérés: 2023. november 27. [Online]. Elérhető: <https://bitbucket.org/product/version-control-software>
- [29] „Azure DevOps Services | Microsoft Azure”. Elérés: 2023. november 27. [Online]. Elérhető: <https://azure.microsoft.com/en-us/products/devops>
- [30] „Features | GitHub · GitHub”. Elérés: 2023. december 2. [Online]. Elérhető: <https://github.com/features>
- [31] „Why GitLab? | GitLab”. Elérés: 2023. december 2. [Online]. Elérhető: <https://about.gitlab.com/why-gitlab/>

- [32] „A Parancssor alapjai | C# Tutorial.hu”. Elérés: 2024. november 11. [Online]. Elérhető: <https://csharptutorial.hu/docs/hellovilag-hellosharp/3-alkalmazaskeszites-alapjai/a-parancssor-alapjai/>
- [33] „What is PowerShell? - PowerShell | Microsoft Learn”. Elérés: 2024. november 11. [Online]. Elérhető: <https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.4>
- [34] „Introducing the Windows PowerShell ISE - PowerShell | Microsoft Learn”. Elérés: 2024. november 11. [Online]. Elérhető: <https://learn.microsoft.com/en-us/powershell/scripting/windows-powershell/ise/introducing-the-windows-powershell-ise?view=powershell-7.4>
- [35] „Business Central Administration Shell - Business Central | Microsoft Learn”. Elérés: 2024. november 11. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/administration/administration-shell>
- [36] „Documentation for Visual Studio Code”. Elérés: 2023. november 27. [Online]. Elérhető: <https://code.visualstudio.com/docs>
- [37] „AL Language extension for Microsoft Dynamics 365 Business Central - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=ms-dynamics-smb.al>
- [38] „AL Extension Pack for VSCode”. Elérés: 2023. november 27. [Online]. Elérhető: <https://www.waldo.be/2019/07/02/al-extension-pack-for-vscode/>
- [39] „AL Variable Helper - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=rasmus.al-var-helper>
- [40] „AZ AL Dev Tools/AL Code Outline - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=andrzejzwierzchowski.al-code-outline>
- [41] „AL Object Designer - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=martonsagi.al-object-designer>
- [42] „AL Toolbox - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=BartPermentier.al-toolbox>
- [43] „Create GUID - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=nwallace.createGUID>
- [44] „Docker extension for Visual Studio Code”. Elérés: 2023. november 27. [Online]. Elérhető: <https://code.visualstudio.com/docs/containers/overview>

- [45] „vscode-icons - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=vscode-icons-team.vscode-icons>
- [46] „GitLens — Git supercharged - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens>
- [47] „Git History - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=donjayamane.githistory>
- [48] „PowerShell - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=ms-vscode.PowerShell>
- [49] „Snippet Creator - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=wware.snippet-creator>
- [50] „REST Client - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=humao.rest-client>
- [51] „TODO Highlight - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=wayou.vscode-todo-highlight>
- [52] „Todo Tree - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=Gruntfuggly.todo-tree>
- [53] „XLIFF Sync - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=rvanbekkum.xliff-sync>
- [54] „AL Test Runner - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=jamespearson.al-test-runner>
- [55] „AL CodeActions - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=davidfeldhoff.al-codeactions>
- [56] „AL Navigator - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=wbrakowski.al-navigator>
- [57] „AL XML Documentation - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=365businessdevelopment.bdev-al-xml-doc>
- [58] „mdAL - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=joneug.mdal>
- [59] „NAB AL Tools - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=nabsolutions.nab-al-tools>

- [60] „AL Object ID Ninja - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=vjeko.vjeko-al-objid>
- [61] „Partial Diff - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=ryu1kn.partial-diff>
- [62] „BusinessCentral.LinterCop - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=StefanMaron.businesscentral-lintercop>
- [63] „Bracket Select - Visual Studio Marketplace”. Elérés: 2023. november 27. [Online]. Elérhető: <https://marketplace.visualstudio.com/items?itemName=chunsen.bracket-select>
- [64] „How to Choose Between Cloud & On-Premises Dynamics 365 Business Central”. Elérés: 2024. november 6. [Online]. Elérhető: <https://community.dynamics.com/blogs/post/?postId=a162d7ca-e8e0-47e4-83af-df90f435a767>
- [65] „Get started with the container sandbox development environment - Business Central | Microsoft Learn”. Elérés: 2024. november 6. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-get-started-container-sandbox>
- [66] „Sandbox environments - Business Central | Microsoft Learn”. Elérés: 2023. november 27. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/admin-sandbox-environments>
- [67] „Production and Sandbox Environments - Business Central | Microsoft Learn”. Elérés: 2023. november 27. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/administration/environment-types>
- [68] „Managing Production and Sandbox Environments - Business Central | Microsoft Learn”. Elérés: 2024. november 5. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/administration/tenant-admin-center-environments>
- [69] „Get started with AL - Business Central | Microsoft Learn”. Elérés: 2023. november 27. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-get-started>
- [70] „Docker overview | Docker Docs”. Elérés: 2023. november 27. [Online]. Elérhető: <https://docs.docker.com/get-started/overview/>
- [71] „Running a container-based development environment" - Business Central | Microsoft Learn”. Elérés: 2023. november 27. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-running-container-development>

- [72] „BcContainerHelper | Freddys blog”. Elérés: 2023. november 27. [Online]. Elérhető: <https://freddysblog.com/2020/08/11/bccontainerhelper/>
- [73] „Install Business Central Using Setup - Business Central | Microsoft Learn”. Elérés: 2024. november 5. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/install-using-setup>
- [74] „How to Choose Between Cloud & On-Premises Dynamics 365 Business Central”. Elérés: 2024. november 19. [Online]. Elérhető: <https://community.dynamics.com/blogs/post/?postid=a162d7ca-e8e0-47e4-83af-df90f435a767>
- [75] „Component and System Topology - Business Central | Microsoft Learn”. Elérés: 2024. november 11. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/product-and-architecture-overview>
- [76] „Cloud vs On-Premise | Dynamics 365 Business Central Pros & Cons”. Elérés: 2023. december 2. [Online]. Elérhető: <https://www.tvisiontech.co.uk/blog/2019/09/13/business-central-on-premise-vs-cloud-saas/>
- [77] „Testing the advanced sample extension - Business Central | Microsoft Learn”. Elérés: 2023. november 27. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-extension-advanced-example-test>
- [78] „Publish-NAVApp (Microsoft.BusinessCentral.Apps.Management) - Dynamics NAV | Microsoft Learn”. Elérés: 2024. november 5. [Online]. Elérhető: <https://learn.microsoft.com/en-us/powershell/module/microsoft.dynamics.nav.apps.management/publish-navapp?view=businesscentral-ps-24>
- [79] „Install-NAVApp (Microsoft.BusinessCentral.Apps.Management) - Dynamics NAV | Microsoft Learn”. Elérés: 2024. november 5. [Online]. Elérhető: <https://learn.microsoft.com/en-us/powershell/module/microsoft.dynamics.nav.apps.management/install-navapp?view=businesscentral-ps-24>
- [80] „What is an API Platform? | API Platform Capabilities”. Elérés: 2024. november 6. [Online]. Elérhető: https://www.softwareag.com/en_corporate/resources/api/article/api-platform.html
- [81] „Postman documentation overview | Postman Learning Center”. Elérés: 2024. november 5. [Online]. Elérhető: <https://learning.postman.com/docs/introduction/overview/>
- [82] „An Introduction to API Testing with SoapUI”. Elérés: 2024. november 6. [Online]. Elérhető: <https://www.soapui.org/getting-started/introduction/>
- [83] „REST API Documentation Tool | Swagger UI”. Elérés: 2024. november 6. [Online]. Elérhető: <https://swagger.io/tools/swagger-ui/>

- [84] „What are Software Development Methodologies | 15 Key Methodologies - GeeksforGeeks”. Elérés: 2024. november 19. [Online]. Elérhető: <https://www.geeksforgeeks.org/what-is-software-development-methodology-15-key-methodologies/>
- [85] „Waterfall Model - Software Engineering - GeeksforGeeks”. Elérés: 2024. november 11. [Online]. Elérhető: <https://www.geeksforgeeks.org/waterfall-model/>
- [86] „What is Agile Methodology? - GeeksforGeeks”. Elérés: 2024. november 11. [Online]. Elérhető: <https://www.geeksforgeeks.org/what-is-agile-methodology/>
- [87] „What is Scrum in Software Development? - GeeksforGeeks”. Elérés: 2024. november 11. [Online]. Elérhető: <https://www.geeksforgeeks.org/scrum-software-development/>
- [88] „What is Kanban - Principles and Implementation - GeeksforGeeks”. Elérés: 2024. november 11. [Online]. Elérhető: <https://www.geeksforgeeks.org/what-is-kanban/>
- [89] „Install a version 24 update - Business Central | Microsoft Learn”. Elérés: 2024. november 11. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/upgrade/upgrading-cumulative-update-v24>
- [90] „JSON files - Business Central | Microsoft Learn”. Elérés: 2023. november 27. [Online]. Elérhető: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-json-files>
- [91] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 19. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/VAT-Definition>
- [92] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 19. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Cash-Register>
- [93] „DSFinV-K API | fiskaly.developer”. Elérés: 2024. november 19. [Online]. Elérhető: <https://developer.fiskaly.com/api/dsfinvk/v1#tag/Cash-Point-Closing>
- [94] „Setup Table | alguidelines.dev - Business Central Design Patterns”. Elérés: 2024. november 11. [Online]. Elérhető: <https://alguidelines.dev/docs/navpatterns/patterns/singleton/singleton-table/setup-table/>

ÁBRAJEGYZÉK

ábra 1: POS rendszerek összehasonlítása[8].....	10
ábra 2: Verziókezelő szoftverek összehasonlítása[8]	16
ábra 3: Parancssorkezelők összehasonlítása	17
ábra 4: Környezetek összehasonlítása[8].....	23
ábra 5: API fejlesztő platformok összehasonlítása[8]	25
ábra 6: Rendszer topológia[75].....	29
ábra 7: Business Central komponensek[89].....	30
ábra 8: LS Central alkalmazás hierarchia[8].....	31
ábra 9:Tranzakció folyamat[8].....	35
ábra 10: User stories[8].....	39
ábra 11: fiskaly dashboard[8].....	41
ábra 12: fiskaly API Kulcs[8]	41
ábra 13: OpenAPI definíció[8].....	42
ábra 14: OpenAPI definíció raw[8].....	42
ábra 15: Postman gyűjtemény[8]	43
ábra 16: SaaSkörnyezet.....	43
ábra 17: Appsource alkalmazások installálása[8]	44
ábra 18: Logic Factory struktúra[8]	45
ábra 19: ITokenService[8].....	45
ábra 20: IAuthentication[8].....	45
ábra 21: IPOSVatCode[8]	46
ábra 22: IVATDefinitionUpdate[8]	47
ábra 23: IStore[8]	47
ábra 24: Cash Register[8].....	48
ábra 25: Cash Point Closing[8].....	49
ábra 26: Service Locator	49
ábra 27: DSFinV-K CPC Update[8].....	50
ábra 28: fiskaly API key teszteléshez[8]	51
ábra 29: Fiskaly Setup[8]	51
ábra 30: Store karton beállítások[8].....	52
ábra 31: fiskaly Dashboard TSS[8].....	52
ábra 32: fiskaly kliens BC[8]	52
ábra 33: fiskaly kliens Dashboard[8]	53
ábra 34: Cash Register beszúrása, frissítése[8].....	53
ábra 35: Cash Register létrehozva BC[8].....	53
ábra 36: Cash Register létrehozva fiskaly[8].....	54
ábra 37: Slave Cash Register[8].....	54
ábra 38: POS VAT Code beállítás[8].....	54
ábra 39: Fiskaly Sales Transaction[8].....	54
ábra 40: Pending Cash Point Closing[8].....	55
ábra 41: Completed Cash Point Closing[8]	55
ábra 42: Completed Cash Point Closing fiskaly[8].....	55

ábra 43: Feladatvárólista beállítás[8]	55
ábra 44: Helyi környezet létrehozása script 1[8]	56
ábra 45: Helyi környezet létrehozása script 2[8]	56
ábra 46: Teszt library installálás 1[8]	56
ábra 47: Teszt library installálás 2[8]	57
ábra 48: Teszt alkalmazás függőségek[8]	57