

1. Bevezetés

1.1 Jelenlegi rendszer bemutatása & tervek

A jelenleg működő és használatban lévő megoldások fő problémája az, hogy egybe van építve a CMS rendszer (mely a webshop tartalmat kezeli) az ERP rendszerrel. Az így létrejött komplex megoldás adatstruktúrája túlmutat az eredeti terveken, ezért lassú és nem moduláris a kód. Az EperFit komplex ERP ezt hivatott modulárisan kezelni, mely modulonként elérhető.

Jelen dokumentációban az *eperfitdemo.com* domain névre fogunk hivatkozni a könnyebb érthetőség kedvéért, azonban természetesen minden ügyfél saját domain nevén éri el az alkalmazást (ez az oldal nem létezik, a tesztelés az *app.eperfit.com* valamint az *eper.fit* domain alól elérhető).

Maga a *eperfitdemo.com* honlap, melyen keresztül az értékesítés zajlik, el kell, hogy különüljön a háttér rendszertől.

A vállalatirányítási rendszer eredményeképpen a jelenleg meglévő centralizált rendszerről leválasztásra kerülnek az alábbiak:

- a. Online értékesítés, azaz webshop funkció (lásd bővebben 2.1)
- b. Készletgazdálkodás modul (lásd bővebben 2.2)
- c. Értékesítés menedzsment modul (lásd bővebben 2.3)
- d. Gyártás menedzsment modul (lásd bővebben 2.4)
- e. Mobil applikáció (lásd bővebben 2.5)

1.2 Piaci kitekintés

A raktárkészlet kezelő szoftverek közül számtalan elérhető létezik. A Megrendelő igénye azonban egy olyan integrált megoldás, melyet nem képes egyetlen már meglévő kiszolgálni.

Legfontosabb szempontok, melyek indokolják az egyedi fejlesztés szükségességét:

- a. Adatbiztonság és 99,99% SLA (amennyiben egy szoftver saját gondozású, úgy harmadik fél felé kisebb a kitétsége, mintha bérelne: sem szerver leállásról, sem beszállítói kitétségtől nem kell tartani)
- b. Magyar, valamint uniós jogszabályi megfelelésség a számlázás és a fizetés területén
- c. Egyedi elemzési lehetőségek a meglévő adatokból, mint árjavaslatok alkalmazása, valamint terméktörténelem szemlélet: ezeket az igényeket jelenleg piaci szoftver nem képes kielégíteni
- d. Ergonomikus, belső folyamatoknak és igényeknek megfelelő kialakítás
- e. Integrációk harmadik felek felé, korlátlan bekapcsolódási lehetőséggel
- f. Időmegtakarítás: mentés és küldés gombok nélküli adatszinkron segítségével el tudjuk érni, hogy az adminisztrációra fordított idő több, mint 50%-al csökkenjen: hiszen nem kell a mentés gombra nyomni, mert a rendszer automatán ment, valamint nem kell a szerkesztés

gombra kattintani, mert az oldalon belüli szerkesztés is lehetővé válik. Ezen között pedig line-option megoldással lehet válogatni, ki mit szeretne nézni és szerkeszteni

1.3 Fejlesztés lépései, követelmények

A fejlesztés során elsődleges szempont, hogy a létrejövő alkalmazás tulajdonjoga és értékesítési joga a megrendelőhöz kerüljön, megfelelő licenzek és terméktámogatás biztosításával.

Javasoljuk, hogy fejlesztés során (külsős vagy saját) követelménymérnök ellenőrizze a fejlesztés szakmai lépéseit, stratégiai döntéseit. Az erre allokált költségkeret 80 órányi követelménymérnöki munkaóra.

Továbbá javasoljuk, hogy a meglévő rendszert fejlesztő kolléga vegyen részt a szakmai előkészítésben és tanácsadásban, ismertesse a jelenlegi program nehézségeit és tipikus hibáit. Az erre allokált költségkeret 30 órányi szakértői munkaóra.

Megvalósítási ütemterv:

- 1) Tervdokumentáció készítése
 - a. Ebben a fázisban elkészül a részletes dokumentáció az ügyféligény felmérésével, valamint a konkurens hasonló programok áttekintésével. Ezeket az összegzéseket jelen dokumentum tartalmazza.
 - b. Követelménymérnöki szempontrendszer felállítása, melyet szintén jelent dokumentum 6. pontja tartalmaz.
- 2) Frontend fejlesztés megvalósítása (2.2 – 2.4 pontok)
- 3) Backend fejlesztés megvalósítása (2.2 – 2.4 pontok)
- 4) API adatcsatornák összekötése a webáruházhoz, illetve harmadik fél szoftvereihez
- 5) Rendszer átadása és élesítése tesztüzemben, majd véglegesen
- 6) Mobil applikáció kiadása és élesítése

1.4 Fejlesztés céljai

A fejlesztés elsődleges célja a selejtek minimalizálása és az átláthatóság biztosítás. A komplex szoftverrel azt kívánjuk elérni, hogy minden egyes árumozgás, pénzmozgás visszakövethető legyen. A rendszer megvéd a csalástól, és egyszerűsíti az adminisztrációt.

Továbbá rendkívül fontos, hogy élőmunkát takarít meg a rendszer. Nem szükséges adminisztrációra külön munkavállalót megbízni.

2. Projekt felépítése, részei

2.1 Online értékesítés

Az értékesítés az egyetlen felület, melyet a külső felek számára nyitott. Klasszikus webáruház rendszer, melyen keresztül az értékesítés zajlik, elérhető az *eperfitdemo.com* oldalon keresztül.

Lehet a webáruház egy bérelhető megoldás, vagy akár a jelenleg meglévő, viszont képesnek kell lennie arra, hogy API adatcsatornán keresztül kommunikáljon az ERP rendszerrel.

Ezt az adatcsatorna kommunikációt biztosítja a nyílt forráskódú Woocommerce, OpenCart, a zárt forráskódú ShopRenter (részleges támogatással), valamint bármilyen egyedi fejlesztés. Egyedi fejlesztés választása esetén javasoljuk a PHP Laravel backend keretrendszert HTML5+CSS3 frontend fejlesztéssel, jQuery alapú AJAX kommunikációval annak érdekében, hogy könnyen keresőoptimalizálható legyen, ugyanakkor a rendszer bővíthetővé váljon bárki hozzáértő számára egyszerűen.

A 2.1 rendszer elérhető a eperfitdemo.com oldalon keresztül.

Az API kommunikáció az api.eperfitdemo.com/api elérés alól történik.

2.2 Készletgazdálkodás modul

A készletgazdálkodás modul a raktárak közötti, és a raktárakon belüli mozgásokat kezeli és menedzseli.

Az árukészlet mozgatása történhet kézzel vagy gépi rögzítéssel. Gépi rögzítés egy olyan esemény, melyet valamely kézi rögzítés generál: azaz egy rendelés leadása API adatcsatornán keresztül, vagy egy rendszeren belüli változás indikálja.

A legfontosabb funkció ebben, és az alábbiakban is, az az oldalon belüli szerkesztés. Azaz nem szükséges tovább kattintania a felhasználónak, elegendő az oldalon belül módosítani valamit, és a rendszer automatikusan ment. Erről visszajelzést küld. Így a hagyományos rendszerekhez képest több, mint 50% időmegtakarítás érhető el.

A 2.2 rendszer elérhető a stock.eperfitdemo.com oldalon keresztül.

Az API kommunikáció az api.stock.eperfitdemo.com/api elérés alól történik.

2.3 Értékesítés modul

Az értékes modulba futnak be a webáruház rendelések, valamint a bolti eladások (terminálból). A rendszer az értékesítés folyamatát kezeli, azonban szoros kapcsolatban van a készletgazdálkodással.

Ez a rendszer felel azért, hogy az összes termék kezelhető legyen.

A 2.3 rendszer elérhető a sales.eperfitdemo.com oldalon keresztül.

Az API kommunikáció az api.sales.eperfitdemo.com/ api elérés alól történik.

2.4 Gyártás menedzsment modul

A gyártás menedzsment modul a 2.3, 2.4 rendszerektől külön álló, azokkal API adatcsatornán összekötött rendszer.

A gyártás menedzsment modulban ugyan úgy termékeket kezelünk, azonban ezek a termékek összetevői a 2.2, 2.3 pontokban definiált rendszernek. Azaz lehetővé válik egy beszállítói megrendelés leadására a készletgazdálkodás funkciói közül, ami a 2.4-es modulban komponenseire bomlik, azaz megnézi, hogy a berendelésre szoruló termékekhez milyen eszközök, összetevők, és anyagok kellene, és azt leadja megrendelésre. Ugyancsak segít abban a rendszer, hogy anyaggazdálkodást végezzen.

A 2.4 rendszer elérhető a prod.eperfitdemo.com oldalon keresztül.

Az API kommunikáció az api.prod.eperfitdemo.com/ api elérés alól történik.

2.5 Mobil applikáció

A mobil applikáció a weben elérhető rendszerrel azonos funkcionalitással fog rendelkezni. Az applikáció készítése a rendszerrel együtt fejlődik, azaz minden funkció először megjelenik a rendszerben, majd kialakításra kerül az applikációban is.

A mobil applikáció fejlesztése során a frontend, azaz a külső megjelenés a rendszer reszponzív verziójával lesz megegyező.

A fejlesztés tartalmazza a közzétételt (és az azokhoz szükséges dokumentációkat, képernyőképeket, ábrákat) is a Google Play Áruház, valamint az AppStore platformjain.

A mobil applikáció egyező API adatcsatornákat használ a webes rendszerrel.

3. Technológiai megvalósítás

3.1 Adatszerkezeti felépítés

Alapvetően megkülönböztetünk két dolgot ennél a pontnál. Külön kezeljük a webshopot, melynek vázlatos technológiai funkcióit tartalmazza a 2.1-es pont. Jelen pont kizárólag a mögöttes rendszer technológiai felépítését tartalmazza. Ezen rendszerek egyazon technológiára fognak épülni.

A modularitás megköveteli nemcsak az adattáblák részletes megtervezését, hanem a különböző modulok szeparált kezelését. Azonban fontos, hogy némely adat duplikáltan kerül kezelésre. Jó példa erre, hogy termékek nyilvántartásra szükség van a webáruház, a készletkezelés, értékesítés, valamint a gyártás rendszerében is, azonban a pontos készlettörténet kizárólag a készletkezelés felületén lesz elérhető.

Cél az adatstruktúra tervezés során, hogy a redundanciát csökkentsük, azonban a modularitást biztosítsuk. Az előzetes egyeztetések során elképzelhető, hogy a b, c és d adatbázisok összevonásra kerülnek, és csak a webshop lesz leválasztva.

A projekt során létre fog jönni több adatbázis:

- a. Az *eperfit_shop* adatbázis felel a webáruház tartalmáért. Ide kerül minden külső, de fontos adat.
- b. Az *eperfit_erp_stock* felel a készletgazdálkodásért.
- c. Az *eperfit_erp_sales* felel az értékesítésért.
- d. Az *eperfit_erp_prod* felel a gyártásért.

3.2 Felhasznált Technológiák – frontend fejlesztéshez

Frontend fejlesztés során a legfrissebb, HTML5 és CSS3 technológiákat alkalmazzuk. Mindemellett a dinamikus tartalmak megjelenítéséhez alkalmazzuk a JavaScript-et, valamint az arra épülő jQuery-t.

Frontend fejlesztés során a Bootstrap keretrendszert használjuk a HTML és CSS alapokhoz, mely számos előre épített CSS és a JavaScript elemet tartalmaz. Ez lehetővé teszi a reszponzív, azaz mobilbarát weboldal és webapplikáció elkészítését úgy, hogy az egy megfelelően hozzáértő fejlesztő által könnyen alakítható legyen a későbbiekben.

A létrejövő frontendet komponensekké alakítjuk a StencilJS felhasználásával, hogy a létrejövő komponensek maximálisan újra felhasználhatóak legyenek.

A létrejövő HTML és CSS elemekből generált StencilJS komponensek megjelenéséért a Vue3 frontend framework felel, így a fejlesztés egy SPA (Single Page Application) lesz eredményében, ami PWA-ként is használható.

A framework felel a dinamikus adatbetöltésért, optimalizációért, mely az SPA-k előnye a nem dinamikus oldalakkal szemben. Így a felhasználó számítógépe csak akkor küld lekérdezést a szerver felé, amikor az szükséges, egyéb esetben a szerver nem terhelődik.

3.3 Felhasznált Technológiák – backend fejlesztéshez

A backend összes eleme egy konténerizált Dockeres környezetben fut. Ez elsődlegesen tartalmaz egy adatbázis kiszolgálót, mely egy MariaDB alapú MySQL.

A backend számítási környezetét a két kiszolgáló szerverre telepített Laravel keretrendszer kommunikációja biztosít. A PHP programnyelv kódja így egy belső API-n keresztül CRUD műveletekkel kommunikálnak a frontend framework-el.

Az API-k REST csatornán, valamint real-time esetekben (chat, értesítések) WebSocket alapon kommunikálnak.

A központi szerver autorizációs tokeneken keresztül kommunikál multimédia tárolószerverrel, ami a dokumentumok, videók és képek tárolásáért felel, így gyorsítva a kiszolgálást.

3.4 Kiszolgáló környezet

A Cloud kiszolgáló partnerünk az ArubaCloud. Tervezett specifikáció a rendszerhez: *ArubaCloud Cloud PRO Debian10 64bit 20GB SSD*

Az ArubaCloud megfelel a legszigorúbb GDPR elvárásoknak, melyről ezen a linken lehet részletesebben olvasni: <https://www.arubacloud.hu/gdpr-adatvedelem-eu-szabalyozas.aspx>

3.5 Multimédia tároló környezet

Tervezünk kialakítani egy multimédiás tároló szervert. (Ennek elérése a *storage.eperfitdemo.com* lesz.) Erre azért van szükség, mert a nagy mennyiségű videós anyagot így elkülöníthetjük. A mindennapi használatban előforduló fájlok a fő szerveren kapnak helyet, hiszen ezek használják a RAM és a CPU erőforrásokat. Azokat a fájlokat, melyeket ideiglenesen kell csak előhívni ezen a külön tárolószerveren kapnak helyet.

Tervezett paraméterei (két opciót javasolunk):

- 1) BASIC 2.3, Atom DC (2× 1.6 GHz) (mag: 2), 4 GB RAM, 500 GB HDD – kiszolgáló az ArubaCloud magyarországi leányvállalata: <https://www.forpsi.hu/serverdedicated/basic/>
- 2) Dell PE R210 II, Xeon E3-1230v2 3,3GHz, 16 GB RAM, 2x1TB SATA – kiszolgáló az ArubaCloud magyarországi leányvállalata: <https://www.forpsi.hu/serverdedicated/value/>

Ezzel az elkülönítéssel egyrészt a főfunkciók betöltését gyorsítjuk, másrészt pedig költséget csökkentünk, hiszen a kevésbé lényeges objektumokat SSD helyett HDD-n tároljuk.

3.6 Támogatott böngészők

Minden modern, legalább ECMA 6 szabvány támogatását biztosító webböngésző támogatott lesz, így a Google Chrome, Firefox, Edge és Opera böngészők legújabb verziói mind támogatást kapnak.

A böngésző támogatás ellenőrzéséhez a Lambdatest szoftvert használjuk, az adattárolás és bejelentkezés, valamint támogók blokkolása során az általános iparági szabványokat követjük, többek között: Apple HMI, Google HMI, OOP.

Asztali számítógépekhez:

- a. Chrome (v84 or above)
- b. Opera (v61 or above)
- c. Firefox (v78 or above)
- d. Safari v13 or above)
- e. MS Edge (v87 or above)

Mobil eszközökhöz:

- a. IOS Safari (iPhone 6 or above)
- b. Android Chrome (Android 7 or above)
- c. Samsung Internet (v12 or above)

3.7 Mobilapp technológiák

A Vue3 framework-ből Capacitor technológián keresztül építjük fel az IOS, Android, valamint igény esetén a desktop-ra telepíthető verziót.

4. Skálázhatóság biztosítása

4.1 Scalability by Design

Alapelvként a fejlesztés során elválasztásra kerülnek a fő rendszerek, valamint az ahhoz kapcsolódó backend és frontend szolgáltatások. Így ezek kommunikáció gördülékeny, támogatásuk és fejlesztésük szeparáltan történhet.

4.2 Plugin megoldás harmadik fél által történő bővítésére

A plugin megoldások segítségével egy olyan rendszert tudunk megkapni, mely lényege, hogy külső fejlesztők bevonásával extra funkciókat képesek megvalósítani, az eredeti fejlesztők felügyelete nélkül. Ennek segítségével nem egy sima ERP, hanem egy generikus platformot kapunk, ami innentől kezdve korlátlanul nőhet, az alap forráskód módosítása nélkül. A rendszer működési logikájának ez az alapja.

Ehhez egy plugin fejlesztési útmutató kerül kialakításra, melynek segítségével bárki elsajátíthatja a rendszer sajátosságait, és későbbiekben hozzájárulhat annak fejlesztéséhez.

A plugin rendelkezik egy telepítőfájllal, ami egy szkript állomány. Ez egyszer képes lefutni, és ezzel fájlokat tud írni a szerverre, valamint az adatbázisba. Lefutása után törlődik, ez egy telepítőfájl, ami az alábbi három típusú egységet képes telepíteni.

A pluginnek jogosultsága a következőkre fog kiterjedni:

- 1) Adattábla bővítés
 - a. A plugin képes lesz arra, hogy az adatbázishoz újabb adattáblát hozzon létre. Ezzel azonban arra nem kap jogosultságot, hogy korábbi adattáblát módosítson, így a felülírás elkerülhető. Az adattábla nem SQL lekérdezéseken keresztül lesz létrehozható. Adattábla beillesztését egy belső függvénnyel tudja a plugin fejlesztője meghívni, hogy védekezzünk az SQL injection ellen.
- 2) Backend API bővítés
 - a. A plugin létrehozhat új API endpointokat, és ehhez CRUD műveleteket.
- 3) Frontend megjelenítés bővítése
 - a. Teljesen új funkció esetén
 - i. Elsődlegesen létrehozhat a plugin a Vue routing script módosításával új tartalmi oldalt. Így létrehozhatunk teljesen új funkciót, ami korábbihoz nem kapcsolódik. *Tipikus példa erre a Facebook Marketing elemzés API.*
 - ii. Jogosultsága van a Vue components mappájához, így hozzáadhat új komponenset. Ez a teljesen új funkció egy részéhez szükséges.
 - iii. Jogosultsága van a Vue views mappájához, így hozzáadhat új tartalmi oldalt. Ebben az esetben leginkább egyetlen fő fájl fog hozzáadódni, ami a komponenseket fogja tartalmazni.

b. Korábbi funkció kiegészítéseként

- i. A view fájlokban fel lehet használni egy `requireExternalComponent` függvényt, ami komponenseket importál, amennyiben létezik. Így ennek segítségével képesek vagyunk egy meglévő view fájlhoz abban előre megadott helyeket új funkciót hozzáadni. *Tipikus példa erre a Letöltés PDF-ként gomb.*

A `requireExternalComponent` függvény lényege, hogy a 3/a/i megoldáshoz hasonlóan importálni tudunk komponenseket a view fájlba. Ha egy meglévő funkciót szeretnénk bővíteni, akkor ennek segítségével tudjuk megtenni, részben pedig üres segédkomponensek meghívásával. Ezt a függvényt üres stringgel tér vissza, ha nem létezik telepített komponens.

Helyek, ahol plugin beillesztési lehetőséget kell adni:

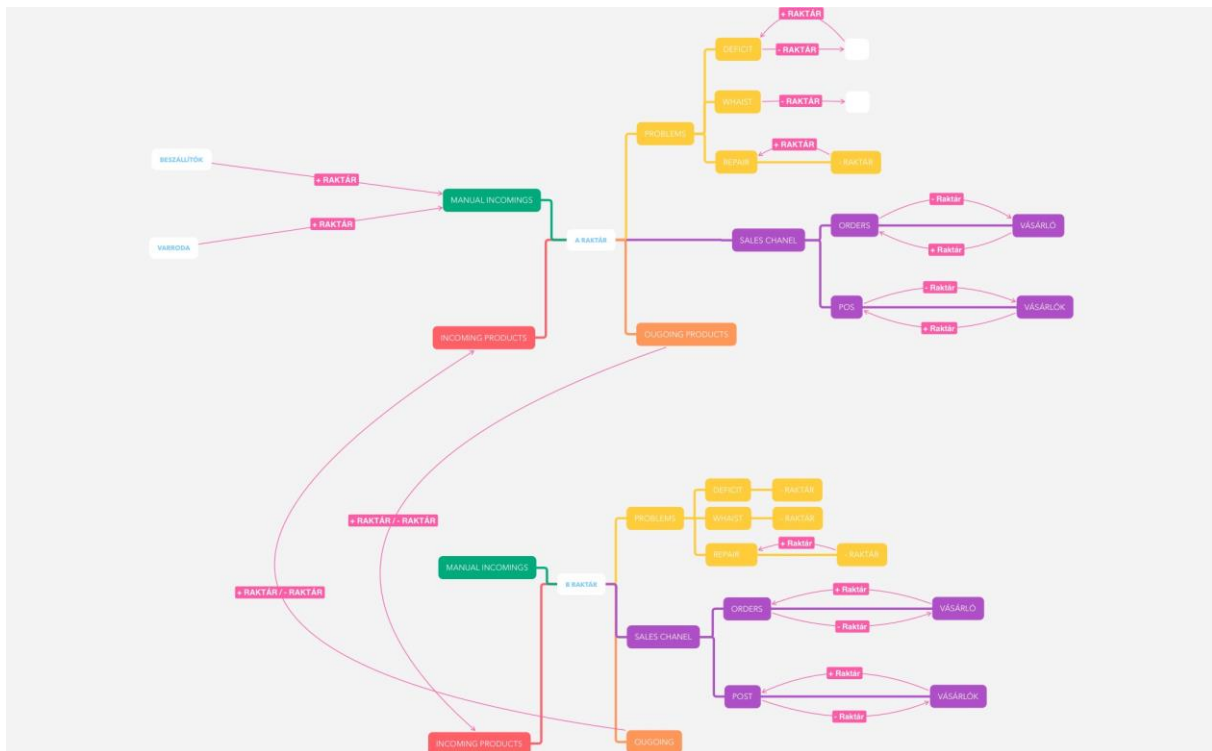
- 1) Fejléc sáv (*Tipikus példa erre gyors funkciókialakítás, mint a Raktárváltó funkció*)
- 2) Tartalmi oldalak teteje, illetve alja (*Tipikus példa erre a gyors összesítőnézet, mint például egy kimutatásdiagram a főoldalra*)
- 3) Táblázatokhoz line option hozzáadási lehetőség – ez a termékekhez egy új opciót ad hozzá, mely így csak adatbázisban rögzül. A frontend pedig opciók alapján renderel, így egyszerűen szerkeszthetővé válik (*Tipikus példa erre egy egyedi tárolni kívánt adatmező*)
- 4) Tartalmi oldalak táblázatainak jobb felső sarka (*Tipikus példa erre a Letöltés PDF-ként gomb*)
- 5) Auto jobok: bizonyos események bekövetkezésekor, mint például adat rögzítés lehetőség lesz egy feladat indikálására. (*Tipikus példa erre az értesítő küldés vagy a számlázás*)

Továbbá lehetőséget kell arra biztosítani, hogy az egyes felhasználók a komponenseket drag'n'drop módszerrel saját igényüknek alakítsák ki. Azaz, ha valakinek az árbevétel a fontos, akkor azt nézi. Ha másnak a készlet, akkor azt helyezi előtérbe. Ugyanígy lehetőséget ad a rendszer arra, hogy bizonyos funkciókat, mely kevésbé jelentős, elrejtünk. Ezek keresőből ugyanúgy elérhetőek, viszont a menüben ilyenkor nem jelenik meg, tényleg csak azt fogja a felhasználó látni, ami számára valóban érdekes is.

5. Funkciók részletezése

5.1 A rendszer alapfunkciói

Alapfunkcióba azok tartoznak bele, melyek a létrejövő rendszer elválaszthatatlan részét képezik, és minden munkavállaló számára fontos.



1. ábra: raktárak közötti árumozgás szemléltetése

A következők tartoznak ide:

- 1) Árumozgatás esetén bevételezés, raktárak közötti átmozgatás, valamint alapvető áruműveletek, mint a selejt, javítás és hiány. Ezzel minden raktáron belüli művelet leírható.
- 2) Terméktörténelem: bankszámla kivonathoz hasonló szemlélet, mely egy kivonatot biztosít a készletváltozásról
- 3) Minimum készletezési beállítás lehetősége
- 4) Teljes felületre kiterjedő funkciókereső, mely a funkciók között, valamint a termékek között tud keresni
- 5) Terméklapon, rendelésoldalon kommentelési lehetőség fájlmegosztással, módosítási történettel (például selejtkezés esetén egy kép feltöltésének lehetősége)
- 6) Leltározás és részleltározás lehetősége
- 7) Várakozási pozícióban lévő termékek: szállításra és bevételezésre várnak
- 8) Termékvariációk és relációk kezelése: adatok összekapcsolásának lehetősége, hogy látható és értelmezhető legyen, mi mihez kapcsolódik

5.2 Alapvető pluginek, melyek fejlesztésre kerülnek

Lesznek olyan fejlesztések, melyek leválasztható modulok (lásd. 4.2 – pluginek). Ezeket a rendszer használója bekapcsolhatja, amennyiben szükséges, de is választhatja, amennyiben már nem tart rá igényt.

A következők tartoznak ide:

- 1) GLS xlsx import beérkezett utánvétek kezelésére
- 2) Szedési lista generálása a raktáros részére, hogy a lehető legrövidebb úton, sorban tudjon haladni
- 3) Vonalkód olvasó programok és vonalkód generálásának lehetősége
- 4) Mutatószámok kalkulálása: a rendszerben rögzített adatokat felhasználva statisztikákat lehet így kinyerni bármiről, amiről szeretnénk: mindenképpen elkészítünk 20-30 darab kimutatást, mint napi/heti/havi fogyasztás, legtöbbet vásárló ügyfél, termelékenységi mutatók.
- 5) Értesítő e-mailek küldése: lehetőség lesz a kommenteléskor betaggelni a felhasználót, valamint megghiúsult fizetésről emlékeztető e-maileket és SMS-t küldeni.
- 6) Feketelistázás funkció: csalásmegelőzésre szolgál, amennyiben egy vevő nem vesz át utánvéteket, a rendszer megjegyzi, és ne fogja neki engedni az utánvételes fizetési lehetőséget.
- 7) Automatikus számlázás integráció, bármilyen kívánt rendszerbe
- 8) Visszárú-selejt-hiba arányok automatikus kalkulálása és kijelzése. Ha kimagaslóan magas az arány (top 3%), akkor a rendszer jelez, hogy az adott termék gyártása álljon le
- 9) Facebook marketing API: ennek segítségével a hirdetési költségek nyomon követhetőek, és látható, hogy termékenként mekkora ráfordítással mekkora eredmény érhető el

Mesterséges intelligencia (vagy ahhoz hasonló) automata elemzési szolgáltatások is pluginként kerülnek be a rendszerbe.

Ezek a következők lesznek:

- 1) Gyártsd: amennyiben fogytán van a termék, javaslatot tesz a gyártásra, illetve az ahhoz kapcsolódó anyagbeszerzésre
- 2) Akciózd: amennyiben termék fogyási görbe felfutó szakaszában van, és a készlet túlságosan nagy, vagy amennyiben akciós időszak van a termék életében árat csökkent a rendszer, szintenként
- 3) Kampányold: amennyiben egy termék megfelelően fogy, de a marketingráfordítása az átlagon aluli, és készleten is van, ezt javasolja
- 4) Kifutó: amennyiben a termékéletciklus végén jár, javasolja a gyártás megszüntetését, és kifutó terméknek jelölést

A fenti négy jelző indikátor a mutatószámokon, és a regressziós modelleken keresztül fogja a terméket elemezni, majd arra javaslatot tenni.

6. Mérnöki követelmények

6.1 Mérnöki követelményrendszer felállítása

Az alábbiakban részletezzük a követelményeket, melyeket a rendszernek tudnia kell. Átadáskor ezek a követelmények kerülnek ellenőrzésre.

1) Look and Feel

QR-101 Az iOS mobil alkalmazás kövesse az Apple HMI iránymutatásait (pl. native iOS HMI controls).

QR-102 Az Android mobil alkalmazás kövesse a Google HMI iránymutatásait (pl. native Android HMI controls).

2) Performance

QR-201 A rendszer egyszerre legalább 1000 db bejelentkezett felhasználót tud kiszolgálni.

QR-202 A rendszer egyszerre legalább 100 db előre rögzített videót néző felhasználót tud kiszolgálni akadásmentes videó lejátszással. Az élővideó teljesítmény-követelményhez lásd a QR-901 feltételt.

QR-203 A rendszer egy lap tartalmat képes legfeljebb 3 s-en belül legenerálni. Megjegyzés: a felhasználó böngészőjében megjelenő tartalom megjelenési ideje sok más tényezőtől függ, például a felhasználó internet kapcsolatának minőségétől.

QR-204 A rendszer egy mobil applikáció képernyő tartalmat képes legfeljebb 3 s-en belül legenerálni. Megjegyzés: a felhasználói felületen megjelenő tartalom megjelenési ideje sok más tényezőtől függ, például a felhasználó internet kapcsolatának minőségétől.

QR-205 A felhasználó web böngészője egy lap tartalmának megjelenítését már akkor megkezdi, mielőtt a teljes tartalom letöltődne.

QR-206 A mobil applikáció egy képernyő tartalmának megjelenítését már akkor megkezdi, mielőtt a teljes tartalom letöltődne.

QR-207 A rendszer üzemen kívüli ideje (MDT) legfeljebb 8,5 óra/év. (99,9%-kal számítva)

QR-208 A rendszer SW frissítése legfeljebb 15 percnyi kiesést jelenthet.

QR-209 Ha egy rendszer művelet mégis nem várt (hibás) állapotba jut, akkor ezt az állapotot a rendszer kezelje le és a rendszer maradjon képes további műveletek indítására (vagyis ilyenkor esetleg egy hibaüzenet jelenhet meg, de a rendszer nem áll le, egy másik művelet a továbbiakban sikeres lehet; persze a sikertelen művelet valószínűleg újra sikertelen lesz).

3) Usability and Humanity

QR-301 A weblap mobil eszközön nem irányítódik át a mobil applikációba.

QR-302 A felhasználó képes a rendszer használatára egy 8 perces képzés elvégzése / video megtekintése után.

QR-303 A weblap képes legyen landscape és portrait módban is megjelenni mobil eszközön.

QR-304 A mobil alkalmazás visszajelez a felhasználónak, amennyiben egy művelet X s-nél tovább tart. Ilyen tipikusan akkor fordul elő, ha az internet kapcsolat minősége alacsony, de akár akkor is, ha a front end lassan válaszol.

QR-305 Az weblap optimalizálva van a böngészők 95%-án.

4) Operational and Environmental

QR-401 Teszt rendszer (szerver) áll rendelkezésre - mely az éles rendszertől független – a rendszer egy új verziójának kipróbálását teszi lehetővé.

5) Maintainability and Support

QR-501 A rendszer skálázható legalább 10x-es mértékig (pl. felhasználó szám, video anyag időtartama) pusztán a HW elemek cseréjével/hozzáadásával.

QR-502 Amennyiben egy rendszer komponensből javított verzió áll rendelkezésre, akkor annak telepítése lehetséges.

QR-503 A Frontend és Backend komponensekhez (különlegesen a Debian, Apache, MySQL, PHP szoftverek) kiadott kritikus frissítések 15 napon belül feltelepülnek a rendszerbe.

QR-504 Naponta biztonsági mentések készülnek (tipikusan MySQL adatbázis) és ebből adatvesztés esetén vissza lehet állítani az adatokat az adatvesztést követő 3. naptári napon.

QR-505 A mobil applikáció a Android 7 és iOS 6 vagy ezek újabb verzióival kompatibilis.

6) Security

QR-601 Érzékeny adatok tárolása: a jelszavak BCrypt kódolással vannak tárolva a MySQL adatbázisban.

QR-602 Legalább egy bizonyos erősségű jelszavak megkövetelése: legalább 8 karakter, kötelező legalább 1 db mindegyikből: kisbetű, nagybetű, szám, speciális karakter. Nincs maximális hossz.

QR-603 SW-es támadások elleni védelem. Az IP cím letiltásra kerül, ha onnan egy órán belül: legalább 50 sikertelen felhasználói belépési próbálkozás történt és a sikeres felhasználói belépések száma kisebb, mint a sikerteleneké.

QR-604 A rendszer a domainhez megbízható DV SSL tanúsítványt biztosít, mely a lejárat előtt megújításra kerül.

7) Compliance

QR-801 A rendszer megfelel a GDPR követelményeknek.

QR-802 A rendszer a bemutatkozó honlapon (TBC) információt biztosít a felhasználónak a cookie használat módjáról.

QR-803 A rendszer a bemutatkozó honlapon nyugtázást kér a felhasználótól a cookie használat módjának megismeréséről.

QR-804 A rendszer a lokalizált szövegeket külön erre a célra fenntartott nyelvfüggő konfigurációs fájlokban tárolja, ahol a lokalizálatlan és lokalizált szövegek egyszerűen párosíthatóak. Egy új nyelv támogatása új konfiguráció hozzáadását jelenti, a nyelvnek megfelelő lokalizált szövegekkel.

8) Extrenal

QR-902 Az ArubaCloud biztosítja a rendszer HW-ének skálázhatóságát legalább 10-szeres mértékig (számítási teljesítmény, tárhely).

7. Jogosultság szerinti felépítés

7.1 Elérhető jogosultsági szintek

Külön felhasználói szintek kerülnek kialakításra, melyeket a következők szerint kívánunk szétosztani:

- 1) Felhasználói hozzáférés: ez az alapértelmezett hozzáférés regisztrációkor, melyet a webshop vásárlók kapnak.
 - a. Lehetőség van regisztráció nélküli vásárlásra, ekkor nem kerül eltárolásra jelszó
 - b. Lehetőség van regisztrációval történő vásárlásra, ekkor eltárolásra kerül egy jelszó. Ennek a segítségével azonosítani tudjuk a felhasználót. A kezelőfelülethez nincsen hozzáférése.
- 2) Megtekintési jogkör, mellyel az alap adatokhoz hozzáfér a felhasználó, de módosítani nem tudja.
- 3) Szerkesztői jogkör, mellyel az alap adatokhoz hozzáfér a felhasználó, és módosítani is tudja azokat.
- 4) Adminisztrátori hozzáférés: ezzel a hozzáféréssel teljes körű statisztikai és exportfunkció elérhető.

A rendszer megkívánja azt, hogy a jogosultságok dinamikusan kezelhetőek legyenek egy menüből. A fentiekén kívül az adminisztrátor által bármikor új jogkör vehető fel, melyhez beállíthatóak, hogy mely oldalakhoz kap megtekintési, illetve módosítási jogosultságot. Ezeket API endpointok alapján tárolja el a rendszer egy jogosultsági táblában.