**Szegedi Tudományegyetem**

Informatikai Intézet

SZAKDOLGOZAT

Somogyvári Hanna

2024

**Szegedi Tudományegyetem**

Informatikai Intézet

Nyugtafelosztó

Szakdolgozat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Készítette: |  | Témavezető: |  |
|  | Somogyvári Hanna |  | Dr. Bilicki Vilmos |  |
|  | programtervező informatikus szakos hallgató |  | egyetemi adjunktus |  |

Szeged

2024

# Feladatkiírás

A szakdolgozat keretein belül egy olyan Angular[[1]](#Angular) alkalmazás elkészítése a cél, ami nyugták végösszegének megadott résztvevők közötti felosztását teszi lehetővé. Emellett képes a feltöltött adatokat tárolni, listázni is.

A felhasználók képesek a vásárlás résztvevőit, és őket tartalmazó csoportokat létrehozni. Ezeket a nyugta egyes termékeihez tudják rendelni, ezt követően az alkalmazás kiszámolja az összeget, amit a termékek árainak megfelelően egy-egy személynek fizetnie kell.

A nyugtákat létre lehet hozni az adatok kézzel való begépelésével, valamint képfeltöltéssel is, ahonnan a szöveget az alkalmazás egy táblázatba illeszti. A táblázat ezután szerkeszthető, valamint Excelbe is kiexportálható.

Az alkalmazás grafikus felhasználói felülete könnyű kezelhetőséget biztosít, és a reszponzív oldalak széleskörű eszközökön való elérhetőséget tesz lehetővé.

# Tartalmi összefoglaló

A téma megnevezése:

Egy Angular[[1]](#Angular) keretrendszerben készült alkalmazás, ami nyugták végösszegének felosztását teszi lehetővé.

A megadott feladat megfogalmazása:

Az alkalmazás egy képként feltöltött vagy manuálisan létrehozott nyugtát képes felosztani résztvevők, csoportok között. Ezt a felhasználó a résztvevők és csoportok előzetes létrehozásával tudja megtenni, akiket utána a nyugta egyes termékeihez rendelheti. Az alkalmazás célja, hogy a megadottak alapján kiszámolja a végösszeget, és az adatokat tárolja.

A megoldási mód:

Az Nyugtafelosztó specifikációja után megterveztem az alkalmazást. Ez a fázis magába foglalja a piackutatást, a diagramok elkészítését és a használt technológiák alaposabb megismerését. A fejlesztést Angular[[1]](#Angular) keretrendszerben valósítottam meg, az adatok tárolásához pedig a Google Firebase[[2]](#GoogleFirebase) szolgáltatást használtam. A folyamatot az alkalmazás ellenőrzésével és tesztelésével zártam.

Alkalmazott eszközök, módszerek:

A diagramokat a draw.io[[3]](#drawio) weboldalon készítettem. A Visual Studio Code[[4]](#VisualStudioCode) kódszerkesztőben fejlesztettem Angular[[1]](#Angular) keretrendszer használatával. Emellett a CSS szintaxisához hasonló, de egyszerűbben használható Sass[[5]](#SCSS) CSS előfeldolgozót alkalmaztam SCSS[[5]](#SCSS) szintaxissal a stíluslapok elkészítéséhez. A GitLab[[6]](#GitLab) platformot használtam egyaránt verziókövetéshez és a fejlesztés feladatainak, mérföldköveinek kezeléséhez. Az adatokat a Google Firebase[[2]](#GoogleFirebase) szolgáltatás segítségével tároltam. A képeken látható szöveg felismerését a Tesseract.js[[7]](#TesseractJs) könyvtárral valósítottam meg.

A fejlesztési folyamathoz az iteratív modellt választottam: először vázlatosan megterveztem az alkalmazást, ezt követően egy-két hetes iterációkra meghatároztam a feladataimat, amiket ez idő alatt részletesebben kidolgoztam és megvalósítottam.

**Elért eredmények:**

Az alkalmazás tervét sikeresen elkészítettem, a piacot felmértem. Mélyebb tudást szereztem az elkészítés során használt eszközökről.

**Kulcsszavak:**

Angular[[1]](#Angular), Google Firebase[[2]](#GoogleFirebase), adatkezelés, szövegfelismerés

# Tartalomjegyzék

[Feladatkiírás 3](#_Toc156852507)

[Tartalmi összefoglaló 4](#_Toc156852508)

[Tartalomjegyzék 5](#_Toc156852509)

[MOTIVÁCIÓ 8](#_Toc156852510)

[1. Területi áttekintés 9](#_Toc156852511)

[1.1. Alkalmazások nyugták tárolásához 10](#_Toc156852512)

[1.1.1. Zoho Expense[8] 10](#_Toc156852513)

[1.1.2. Verify[9] 10](#_Toc156852514)

[1.1.3. Expensify[10] 10](#_Toc156852515)

[1.1.4. Shoeboxed[11] 11](#_Toc156852516)

[1.1.5. Dext Prepare[12] 11](#_Toc156852517)

[1.1.6. Invoice Maker[13] 11](#_Toc156852518)

[1.2. Alkalmazások nyugták felosztásához 12](#_Toc156852519)

[1.2.1 EveryoneAllin[14] 12](#_Toc156852520)

[1.2.2 Best Bill Splitter[15] 12](#_Toc156852521)

[1.2.3. Kittysplit[16] 13](#_Toc156852522)

[2. FUNKCIONÁLIS SPECIFIKÁCIÓ 14](#_Toc156852523)

[2.1. Autentikáció 15](#_Toc156852524)

[2.1.1. Regisztráció 15](#_Toc156852525)

[1.2.1. Bejelentkezés 15](#_Toc156852526)

[2.2. Profil 15](#_Toc156852527)

[2.3. Résztvevők 16](#_Toc156852528)

[2.3.1. Résztvevők listázása 16](#_Toc156852529)

[2.3.2. Résztvevők létrehozása 16](#_Toc156852530)

[2.3.3. Résztvevők módosítása, törlése 16](#_Toc156852531)

[2.4. Csoportok 16](#_Toc156852532)

[2.4.1. Csoportok listázása 17](#_Toc156852533)

[2.4.2. Csoportok létrehozása 17](#_Toc156852534)

[2.4.3. Csoportok módosítása, törlése 17](#_Toc156852535)

[2.5. Nyugták 17](#_Toc156852536)

[2.5.1. Nyugták listázása 18](#_Toc156852537)

[2.6. Nyugták létrehozása 18](#_Toc156852538)

[2.6.1. Csoportok, résztvevők megadása 18](#_Toc156852539)

[2.6.2. Termékek hozzáadása 18](#_Toc156852540)

[2.6.3. Résztvevő, csoport termékhez társítása 18](#_Toc156852541)

[2.6.4. Fejenkénti végösszeg kiszámítása 19](#_Toc156852542)

[2.6.5 Kiexportálás 19](#_Toc156852543)

[2.7. Nyugták szerkesztése, törlése 19](#_Toc156852544)

[3. FELHASZNÁLT TECHNOLÓGIÁK 20](#_Toc156852545)

[3.1. Angular[1] 20](#_Toc156852546)

[3.1.1. Angular Material[19] 20](#_Toc156852547)

[3.1.2. AngularFire[20] 20](#_Toc156852548)

[3.2. Google Firebase[2] 21](#_Toc156852549)

[3.2.1. Authentication 21](#_Toc156852550)

[3.2.2. Cloud Firestore 21](#_Toc156852551)

[3.2.3. Cloud Storage 21](#_Toc156852552)

[3.2.4. Hosting 21](#_Toc156852553)

[3.3. Tesseract.js[7] 21](#_Toc156852554)

[3.4. Sass[5] 22](#_Toc156852555)

[3.5. Egyebek 22](#_Toc156852556)

[Irodalomjegyzék 23](#_Toc156852557)

[Nyilatkozat 24](#_Toc156852558)

[Köszönetnyilvánítás 25](#_Toc156852559)

# MOTIVÁCIÓ

Többszemélyes bevásárlások során, amikor a boltban egy személy fizet, és az összeget a résztvevők utólag osztják fel egymás között, gyakran okoz problémát az egy-egy főre jutó végösszeg kiszámítása. A nyugták rendezetlenek és hosszúra nyúlhatnak, ami hosszas fejtörést okoz, főleg, ha bizonyos termékek csak a csoport egy kisebb részéhez tartoznak. A papíralapú blokkok elkeveredhetnek, a kézzel végzett számításokba akár hiba is kerülhet.

Erre, és még sok hasonló problémára nyújt megoldást a Nyugtafelosztó. Az alkalmazás a blokkról feltöltött képről olvassa ki az adatokat, így nem szükséges azokat kézzel begépelni. Az így elkészült listaelemek mellé a felhasználó hozzárendelheti a vásárlás egyes résztvevőit.

Csoportok létrehozására is van lehetőség, amire leginkább akkor lehet szükség, ha a résztvevők több termékért csak bizonyos személyek fizetnek. Például ilyen lehet, amikor tofut csak a vegetáriánusok, gluténmentes kenyeret pedig csak a lisztérzékenyek vettek. A csoportoknak köszönhetően a felhasználónak nem szükséges minden esetben felsorolni az összes személyt, aki az adott termékért fizet, így gyorsabban és egyszerűbben tudja elvégezni a felosztást.

A Nyugtafelosztó ezek után kiszámítja a személyenkénti végösszeget, így megkönnyíti a nyugta felosztását, lerövidíti az ezzel töltött időt, és megszünteti a kézi számolásból eredő hibákat.

# 1. Területi áttekintés

A piackutatás során a célom olyan alkalmazások megtalálása volt, amelyek kapcsolódnak a Nyugtafelosztó fontosabb funkcióihoz. Az elsődleges szempont a nyugták tárolása, valamint a végösszeg kiszámítása és felosztása volt. Emellett szövegfelismerésre és résztvevők hozzáadására lehetőséget adó alkalmazásokat is kerestem.

A talált alkalmazásokat az eredmények alapján két csoportra osztottam. Az első tagjai a feltöltött nyugták tárolására fókuszálnak, ide tartozik a Verify[[9]](#Verify) és a Shoeboxed[[11]](#Shoeboxed) alkalmazás. A második csoportba pedig azokat az oldalakat soroltam, amelyek célja a végösszeg felosztása, mint például az EveryoneAllin[[14]](#EveryoneAllin) és a Best Bill Splitter[[15]](#BestBillSplitter).

A kutatás során felmért alkalmazások teljes listáját és funkcióit az 1.1. ábra mutatja. Ezeknek az alkalmazásoknak köszönhetően sikerült felmérnem, hogy a Nyugtafelosztónak milyen felhasználói igényeket kell kielégítenie. Ötleteket adtak és példát mutattak a tervezett funkciók megvalósításához. Arra is rávilágítottak, hogy milyen hiányosságokat tudok orvosolni a fejlesztés során, és teljesebb képet kaptam az elérendő céljaimhoz.

A képen szöveg, képernyőkép, szám, tér látható

Automatikusan generált leírás

1.1. ábra – Piackutatás

## 1.1. Alkalmazások nyugták tárolásához

A Nyugtafelosztó főbb funkciói közé tartozik a nyugták szkennelése, és az így keletkezett adatok tárolása, listázása. A piackutatás során több olyan alkalmazást is találtam, amelyek ezeket megvalósítják.

A nyugták feltöltésére több különböző megvalósítást felfedeztem, és olyan példákat is láttam, ahol a digitalizált számla kiexportálására is lehetőség volt.

## 1.1.1. Zoho Expense[[8]](#ZohoExpenses)

A Zoho Expense nem csupán nyugták kezelésével foglalkozik, de mivel ez áll közel a szakdolgozatom témájához, csak az ehhez kapcsolódó funkcióit fogom részletezni. Az alkalmazás használatáért fizetni kell, de ingyenesen kipróbálható 14 napra, valamit demó is igényelhető.

Többféle lehetőség is van a nyugták feltöltésére: beszkennelhetőek telefon kamerájával, e-mailben továbbíthatók az alkalmazásra, valamint felhőről vagy a fájlkezelőből is feltölthetőek. A Nyugtafelosztóban csak a fájlként feltöltést tervezem megvalósítani, de a Zoho Expense[[8]](#ZohoExpenses) megmutatta, hogy nem csak ez az egy mód létezik.

A feltöltött nyugtákat fel lehet osztani, de nem résztvevők hozzáadásával, hanem például a százalék vagy a vásárlás napjának megadásával.

## 1.1.2. Verify[[9]](#Verify)

Ennek az alkalmazásnak az egyik erőssége a képi szövegfelismerés. A feltöltött nyugtáról az adatokat egy táblázatba rendezi, ami szerkeszthető, elmenthető, és később megtekinthető. A beolvasott információk kiexportálására is lehetőséget ad, JSON vagy OCR szöveg formátumban.

## 1.1.3. Expensify[[10]](#Expensify)

Az Expensify[[10]](#Expensify), a Nyugtafelosztóhoz hasonlóan, lehetőséget ad képek szkennelésére és manuális létrehozásra. Viszont az alkalmazás hátránya, hogy csak a nyugta nevét és végösszegét tárolja, valamint a szövegfelismerés sok időt vesz igénybe.

## 1.1.4. Shoeboxed[[11]](#Shoeboxed)

Az egyik legérdekesebb ebben az alkalmazásban, hogy a nyugták hagyományos feltöltési módjai (telefonos szkennelés, e-mail, fájlfeltöltés) mellett a felhasználók elpostázhatják őket a Shoeboxed[[11]](#Shoeboxed) létesítményéhet, ahol az alkalmazottak szkennelik be őket.

## 1.1.5. Dext Prepare[[12]](#DextPrepare)

A Dext Prepare[[12]](#DextPrepare) is képes képről felismerni a nyugta szövegét, legyen az telefonról, e-mailből vagy fájlként feltöltött. Ezután az alkalmazás kiszűri a lényeges információt, és feltölti a rendszerbe. Az adatokat 7 évig tárolják, bankszintű titkosítással.

Arra is lehetőséget nyújt, hogy a felhasználó szinkronizálja a Dext Prepare[[12]](#DextPrepare)-t másik könyvelési alkalmazásokkal, mint az IRIS KashFlow[[17]](#IRIS) és xero[[18]](#xero). Így az itt tárolt adatok a szinkronizált alkalmazásokra is automatikusan feltöltésre kerülnek.

A nyugtákból készült riportokat ki is lehet exportálni, CSV és PDF formátumban, valamint e-mailben is továbbíthatók.

## 1.1.6. Invoice Maker[[13]](#InvoiceMaker)

Az Invoice Maker[[13]](#InvoiceMaker), mint ahogy arra a neve is utal, főként nyugták készítésével foglalkozik. A Nyugtafelosztóval ellentétben itt még nem létezik nyugta, amivel számolni kell, hanem a felhasználó készíti el őket az ügyfelei számára.

A különböző célok ellenére a két alkalmazásban találhatók hasonlóságok, mégpedig az oldal Incvoice Calculator[[13]](#InvoiceMaker) részén. Itt manuálisan lehet megadni a nyugtán szereplő termékeket, ami alapján a végösszeg kiszámításra kerül.

Ezután az elkészített nyugta továbbítható az Invoice Generatorra[[13]](#InvoiceMaker), ami segítségével az adatok megszokott nyugta formátumba szerkeszthetők, például cím, dátum és logó megadásával. A folyamat leegyszerűsítése érdekében az alkalmazás személyre szabható sablonokat is biztosít a nyugtákhoz, amiket a felhasználó letölthet, többek között Word dokumentumként vagy Excel táblázatként. Az elkészült, formázott nyugtát PDF formátumban szintén le lehet tölteni.

## 1.2. Alkalmazások nyugták felosztásához

Több olyan alkalmazás is létezik, ahol a nyugtát egyenlő részekre vagy bizonyos arányban lehet elosztani egymás között. A piackutatás során viszont az volt a célom, hogy a Nyugtafelosztóhoz hasonló funkcióval rendelkező alkalmazásokat keressek, ahol résztvevők hozzáadására is van lehetőség, akik között az alapján oszlik meg a végösszeg, hogy ki melyik terméket vásárolta.

Sajnos a talált alkalmazások között a résztvevők elmentésére egyik esetében sincs lehetőség, ahogy csoportokat sem lehet létrehozni. Ennek ellenére a résztvevők ???

## 1.2.1 EveryoneAllin[[14]](#EveryoneAllin)

Az EveryoneAllin[[14]](#EveryoneAllin) egyszerű felhasználói felületet nyújt, ahol egy nyugta könnyen felosztható a résztvevők között. A számolás során az első lépés a résztvevők létrehozása, majd ezután a termékek hozzáadása következik, egyesével. Itt meg kell adni a termék árát és nevét, hogy ki fizetett érte, és kinek kell fizetnie, ahol kijelöléssel több személy is kiválasztható.

A felhasználó választhat, hogy csoportonként vagy személyenként ossza fel az alkalmazás az adott termék árát. Az előbbi esetében az árat a megadott résztvevők között osztja fel, míg az utóbbi során minden résztvevő a megadott összeget fizeti.

Miután a termékek hozzáadásra kerültek, a felhasználó megtekintheti a teljes számlát, ahol egy lista résztvevőnként mutatja, hogy kinek mennyit kell fizetnie, esetleg mennyivel tartoznak neki. Az alkalmazás hátránya, hogy itt nem jelzi, ki kinek tartozik, ami problémát jelent, ha többen is fizettek.

Az EveryoneAllin[[14]](#EveryoneAllin) telefonos alkalmazással is rendelkezik, ahol a résztvevőket nem csak kézzel lehet létrehozni, hanem más alkalmazásokból is meg lehet hívni őket, mint például a WhatsApp vagy az elmentett telefonszámok használatával.

## 1.2.2 Best Bill Splitter[[15]](#BestBillSplitter)

A Best Bill Splitter[[15]](#BestBillSplitter) több funkciót kínál a nyugták felosztására, és a korábban létrehozott adatok megtekintésére is lehetőséget nyújt. Ehhez a nyugták készítésekor nincs külön mentés gomb, minden automatikusan tárolásra kerül. Egyesével nem lehet törölni ezeket az elmentett nyugtákat, csupán az egész előzmény eltávolítására van lehetőség.

A termékeket egyesével, kézzel kell megadni. Ide lehet képet feltölteni a nyugtáról, de az csak adatként tárolódik, az alkalmazás nem ismeri fel rajta a szöveget.

Résztvevőket termékek létrehozása előtt, de akár közben is hozzáadhat a felhasználó. Ezután az alkalmazás megkérdezi, hogy ki fizetett az adott termékért, itt több résztvevő is megadható, akár különböző összeggel. Az ár elosztása során beállítható, hogy egyenlő félben, megadott százalék szerint, vagy fix értékben számítsa ki a személyenkénti összeget.

Az alkalmazás ezek után kiírja, hogy ki kinek mennyivel tartozik. Az adatok szövegként is kimásolhatók, valamint exportálhatók CSV formátumban. Az eredményt linkként is meg lehet osztani, ahol beállítható, hogy a megtekinthetők csak olvashatják vagy szerkeszthetik is a nyugtát.

## 1.2.3. Kittysplit[[16]](#Kittysplit)

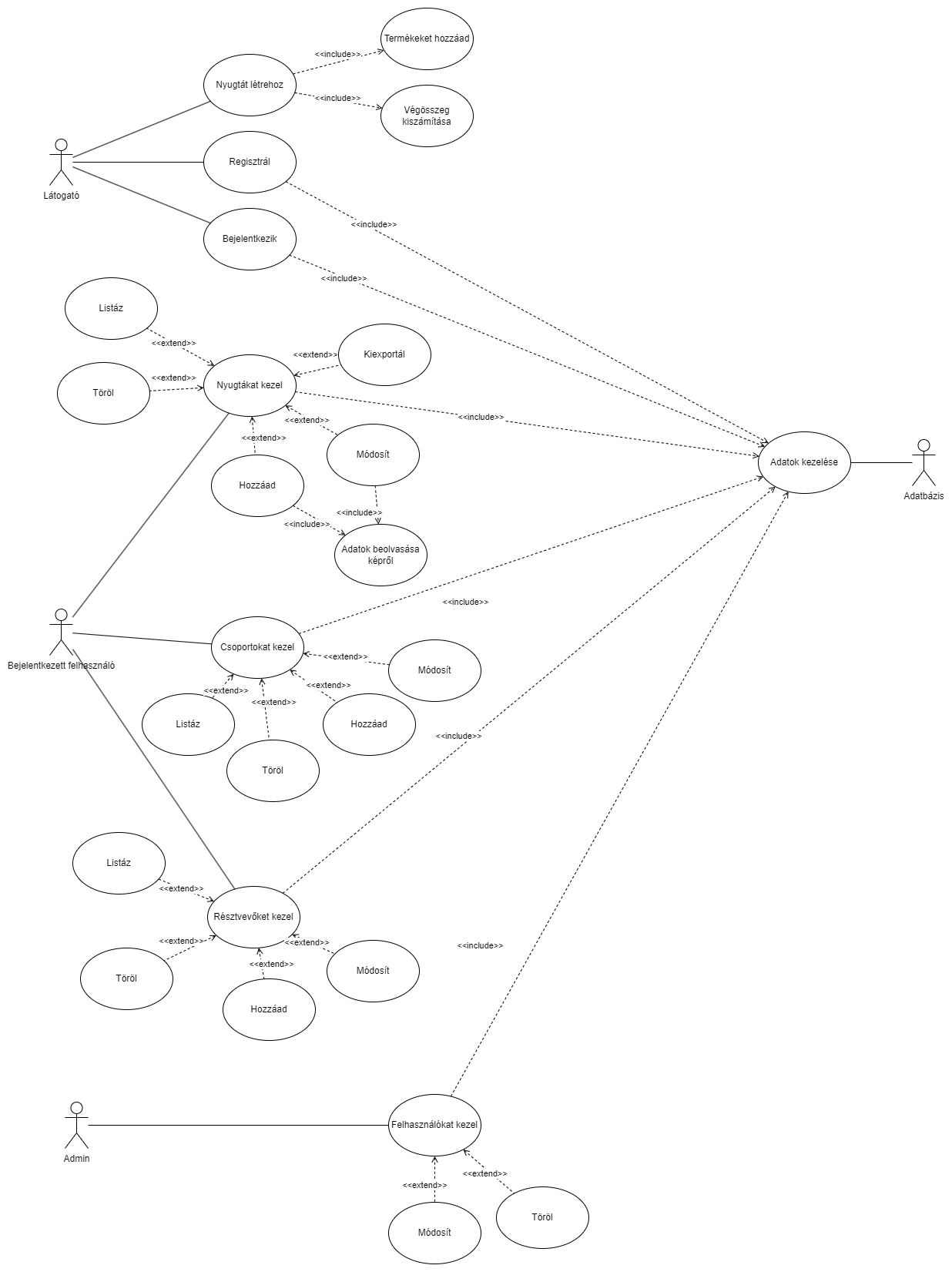
A Kittysplit[[16]](#Kittysplit), a Best Bill Splitterhez[[15]](#BestBillSplitter) hasonlóan, automatikusan elmenti a nyugtákat, amiket a felhasználók létrehoztak. Ezek a főoldalon listázásra kerülnek, és egyenként törölhetőek.

Egy nyugta létrehozásához először meg kell adni a résztvevőket. Ezután a termékeket egyesével lehet hozzáadni, ahol ki kell választani a résztvevőt, aki fizetett, és beírni az árat. Alapértelmezetten az összeg egyenlően kerül elosztásra mindenki között, de van lehetőség másik módot is választani. Az utóbbi esetben az ár elosztható egyenlően a kijelölt résztvevők között is, de van lehetőség százalék vagy fix összeg szerinti elosztásra is.

Az alkalmazás mutatja, hogy ki kinek mennyivel tartozik. A nyugtát linkként meg is lehet osztani, aminek megnyitását követően ki kell választani, hogy a résztvevők közül kik vagyunk. A válaszunk alapján kijelez egy személyre szabott összegzést a nyugtáról, a teljes összeggel, a mi részünkkel, és hogy ebből mennyivel tartozunk még. A link által a nyugta az ingyenes verzióban szerkeszthető is.

# 2. FUNKCIONÁLIS SPECIFIKÁCIÓ

A 2.1. ábrán látható az alkalmazás Use Case diagramja. Az alkalmazás felhasználói három csoportba tartoznak: látogató, bejelentkezett felhasználó és admin. Ezek a csoportok más-más funkciókat érnek el, különböző jogosultsági szinttel. A funkciókat ebben a fejezetben fejtem ki, valamint a 2.1. ábrán is megtekinthetők.



2.1. ábra – Use Case diagram

## 2.1. Autentikáció

Az alkalmazás használatához nem kötelező bejelentkezni, viszont a legtöbb funkció enélkül nem elérhető. Az autentikációs felületen van lehetőség regisztrációra és bejelentkezésre, amit a Google Firebase[[2]](#GoogleFirebase) Authentication szolgáltatással valósítottam meg.

## 2.1.1. Regisztráció

Fiók létrehozásához egy e-mail cím és egy jelszó megadása kötelező. A jelszónak legalább 6 karakter hosszúnak kell lennie, és egy szem ikonra kattintva megjeleníthető, ha a felhasználó leellenőrizné. E-mail cím esetében ellenőrzésre kerül a megfelelő formátum, és a regisztrálás megpróbálása után a felhasználó arról is tájékoztatást kap, ha a megadott cím már foglalt. Sikeres regisztráció esetében egy felugró ablakban jelenik meg erről az információ.

A regisztrált felhasználók a Cloud Firestore[[2]](#GoogleFirebase) adatbázisban is tárolásra kerülnek.

## 1.2.1. Bejelentkezés

A bejelentkezéshez előzetes regisztráció szükséges. Csak regisztrált e-mail címmel és a hozzá tartozó jelszóval lehet bejelentkezni, ha valamelyik helytelen, arról egy hibaüzenet jelenik meg. A jelszó itt is megtekinthető a szem ikon használatával.

Az oldalon egy „Emlékezz rám!” jelölőnégyzet is található, aminek kipipálásával az oldal elmenti a megadott e-mail címet a local storage-ba. A bejelentkezési felület következő meglátogatása során az így tárolt e-mail címet jeleníti meg a mezőben.

Bejelentkezés után az alkalmazás a főoldalra navigál, megváltozik a fejléc, és újabb funkciók válnak elérhetővé. A bejelentkezett felhasználó is a local storage-ban tárolódik, ahonnan kijelentkezés esetén törlésre kerül.

## 2.2. Profil

Bejelentkezett felhasználók képesek a saját profiluk megtekintésére. Itt megjelenik az e-mail címük, és a jelszavuk megváltoztatására is lehetőségük van.

A profil felületről törölhető egy felhasználó fiókja, amit egy megerősítő felugró ablak előz meg.

## 2.3. Résztvevők

A résztvevők olyan személyeket takarnak, akik egy vásárlásban részt vettek, és az összeget közöttük szeretnénk szétosztani. Ezeket a résztvevőket egy nyugta feltöltése előtt létre kell hozni.

## 2.3.1. Résztvevők listázása

A résztvevők nevei egy listában jelennek meg, ami alap esetben ABC szerint rendezett. Ebben a listában csak az adott felhasználó által létrehozott résztvevők láthatók, más felhasználó résztvevői nem érhetőek el. Az adminisztrátor az egyetlen, aki minden adatot elér.

## 2.3.2. Résztvevők létrehozása

Résztvevőket a lista fölött található űrlap segítségével lehet létrehozni. A folyamat felgyorsítása érdekében itt csak a résztvevő neve adható meg, további adatok később adhatók hozzá.

A résztvevők neve nem egyezhet. Ha mégis így történne, arról egy hibaüzenet tájékoztatja a felhasználót.

## 2.3.3. Résztvevők módosítása, törlése

Módosításra és törlésre a résztvevők részletes oldala ad lehetőséget. Ez a listában a résztvevő nevére kattintva érhető el. Ha a felhasználó az URL módosításával egy olyan résztvevőt próbálna elérni, amihez nem fér hozzá vagy nem létezik, az alkalmazás visszanavigál a résztvevők listájához.

A részletes oldalon a résztvevők neve módosítható, és egyéb adatok is megadhatók róluk, mint például a bankszámlaszám.

Törölni is itt lehet őket, az oldalon megjelenő gomb segítségével. Ha egy olyan résztvevő kerül törlésre, aki szerepel egy csoportban, a csoport tagjai közül is el lesz távolítva.

## 2.4. Csoportok

A csoportok több résztvevőt tartalmaznak, és azt a célt szolgálják, hogy felgyorsítsák a nyugta felosztását. Erre leginkább akkor lehet szükség, ha több termékért többször ugyanazok a személyek együtt fizetnek. Csoport létrehozása után ezeket a személyeket nem kell mindegyik terméknél felsorolni, elég csak a csoportot megadni.

Csoportok lehetnek például speciális étrendű vagy valamilyen érzékenységgel rendelkező emberek, mint például a vegetáriánusok vagy a gluténérzékenyek.

## 2.4.1. Csoportok listázása

Csoportok a résztvevőkhöz hasonlóan, ABC sorrendben kerülnek listázásra. Csak az adott felhasználó által létrehozott csoportok láthatóak, kivéve az adminisztrátor esetében, aki minden adatot elér.

## 2.4.2. Csoportok létrehozása

A lista felett látható „Új” gomb megnyomásával hozható létre új csoport. Ekkor egy felugró ablakban meg kell adni a csoport nevét, ami akár tagok nélkül is elmenthető.

Az „Új tag” gomb megnyomása után egy új mező jelenik meg az űrlapon, ahol a csoport egy tagja állítható be. A mező értékét egy lenyíló menüből kell választani, ahol az összes létrehozott résztvevő látható. Egy résztvevő a mellette található x gombbal távolítható el a csoportból.

Csoport létrehozása során minden résztvevő csak egyszer kerül a tagok közé, még abban az esetben is, ha a felhasználó többször adta meg. Az üresen hagyott tag mezőket az alkalmazás figyelmen kívül hagyja. Ha a létrehozni kívánt csoport neve már foglalt, egy hibaüzenet jelenik meg.

## 2.4.3. Csoportok módosítása, törlése

Egy csoportot törölni vagy módosítani annak részletes oldalán van lehetőség. A felületen a csoport neve és egy listában annak tagjai láthatók.

Szerkesztés esetén a létrehozáskor megjelenő ablak jelenik meg, ahol a már ismert módon módosíthatók a csoport adatai. Törlés a részletes oldalon látható gomb segítségével lehetséges.

## 2.5. Nyugták

A nyugták termékek listájából állnak, amik névvel és árral rendelkeznek. Ezekhez a termékekhez felhasználók és csoportok rendelhetők. A Nyugtafelosztó ezek alapján kiszámolja, hogy a résztvevőknek egyenként mennyit kell fizetniük.

## 2.5.1. Nyugták listázása

A felhasználó által létrehozott nyugták ABC sorrendben jelennek meg. Lehetőség van keresésre név, valamint szűrésre bolt és dátum alapján.

## 2.6. Nyugták létrehozása

A lista felett egy „Új” gomb található, aminek megnyomása egy új oldalra vezet. Itt megadható a nyugta neve, dátuma és a bolt, ahol a vásárlás történt.

## 2.6.1. Csoportok, résztvevők megadása

Egy nyugtához tetszőleges mennyiségű résztvevő hozzáadható, mindegyik csak egyszer. A résztvevők egy lenyíló listából választhatók ki, és a mellettük szereplő x-el törölhetők, ha véletlenül lettek hozzáadva.

Egy csoport automatikusan hozzáadásra kerül, ha abból legalább kettő tag szerepel már a nyugtán. Olyan csoport hozzáadására nincs lehetőség, aminek egyik tagja sem résztvevő a nyugtán, de ha csak egy tagja szerepel, akkor manuálisan hozzáadható. Ha egy olyan csoport került hozzáadásra, amit a felhasználó mégsem szeretne használni, azt eltávolíthatja.

## 2.6.2. Termékek hozzáadása

A termékek hozzáadására a Nyugtafelosztó két lehetőséget ad. Az első esetben a felhasználó feltölthet egy képet a nyugtáról, ahonnan az alkalmazás képi szövegfelismerés segítségével egy táblázatba írja a termékek adatait. Ezek az adatok a szkennelés után kézzel módosíthatók.

Második lehetőségként a nyugta manuálisan is létrehozható. Ekkor a felhasználó az „Új termék” gombbal adhat hozzá új sort a táblázathoz, ahová a termék nevét és árát kell beírni. A táblázat alján a teljes végösszeg folyamatosan frissül az aktuális adatok alapján.

Termékek törlésére is lehetőség van a mellettük található x gomb segítségével.

## 2.6.3. Résztvevő, csoport termékhez társítása

A táblázatban minden termék mellett található egy mező, ahová azok a résztvevők vagy csoportok kerülnek, akik az adott termékért fizetnek. Több résztvevő vagy csoport megadható, vegyesen is, amiket egy lenyíló listából lehet kiválasztani, ahol csak a nyugtához hozzáadott csoportok és résztvevők láthatók. Ha valamelyik véletlenül lett hozzáadva, a mellette látható x gomb segítségével törölhető.

Ha egy résztvevő vagy csoport többször kerül megadásra ugyanahhoz a termékhez, az alkalmazás csak egyszer fogja figyelembe venni. Ugyanez lép fel akkor, ha a felhasználó megad egy résztvevőt egy olyan csoport mellé, aminek tagja.

## 2.6.4. Fejenkénti végösszeg kiszámítása

Egy termék ára egyenlően kerül elosztásra a megadott résztvevők és csoportok tagjai között. Az egyes termékek felosztása után az alkalmazás az így kapott értékeket összegzi, és a nyugta alján résztvevőnként megjeleníti.

## 2.6.5 Kiexportálás

A nyugta kiexportálható Excel táblázatba CSV formátumban. Az így kapott táblázatban a nyugta alapadatai, a termékek neve, ára és a hozzájuk rendelt résztvevők (csoportok résztvevőkre bontva), valamint a teljes végösszeg és a kiszámított fejenkénti végösszegek szerepelnek.

## 2.7. Nyugták szerkesztése, törlése

A listában egy nyugtára kattintva a létrehozásról ismert táblázat jelenik meg. Itt a nyugta létrehozásával megegyező módon lehet módosítani, majd elmenteni az adatokat. Törlésre is itt van lehetőség.

# 3. FELHASZNÁLT TECHNOLÓGIÁK

## 3.1. Angular[[1]](#Angular)

Az Angular[[1]](file:///C:\\Users\\hanna\\OneDrive\\Dokumentumok\\Szakdoga\\nyugtafeloszto\\szakdolgozat\\SomogyváriHanna_GLXGEU_szakdolgozat.docx" \l "Angular) egy webfejlesztési, nyílt forráskódú keretrendszer, ami a TypeScript nyelven alapul. A fő célja single-page (egy oldalas) alkalmazások fejlesztése, ahol új weblapok betöltése helyett egyetlen oldal megjelenő tartalmát frissíti dinamikusan a web szerveren tárolt adatokkal.

A keretrendszer többek között egy struktúrát is biztosít az alkalmazások elkészítéséhez. Ennek alapjai a komponensek, amik a rendszer kisebb elemei, mint egy lista vagy egy bejelentkezési űrlap. Egy komponens egy TypeScript osztályból és egy HTML fájlból áll, előbbi a működést, utóbbi pedig a megjelenést definiálja. Szervizek létrehozására is van lehetőség, amik megjelenítéshez közvetlenül nem kapcsolódó funkciókat valósítanak meg, például adatbáziselemekkel való interakciókat. Egy szervizt több komponens is használhat, amit előtte injektálni kell. A szorosan összekapcsolódó komponensek, szervizek és egyéb kódfájlok modulokba szervezhetők.

Fejlesztés során az Angular[[1]](file:///C:\\Users\\hanna\\OneDrive\\Dokumentumok\\Szakdoga\\nyugtafeloszto\\szakdolgozat\\SomogyváriHanna_GLXGEU_szakdolgozat.docx" \l "Angular) 16-os verziójával dolgoztam. Emellett az Angular[[1]](file:///C:\\Users\\hanna\\OneDrive\\Dokumentumok\\Szakdoga\\nyugtafeloszto\\szakdolgozat\\SomogyváriHanna_GLXGEU_szakdolgozat.docx" \l "Angular) lentebb részletezett segédkönyvtárait is használtam.

## 3.1.1. Angular Material[[19]](#AngularMaterial)

Az Angular Material[[19]](#AngularMaterial) egy komponenseket tartalmazó könyvtár. Weboldalak egyszerűbb, gyakran használt elemeiből áll, mint gombok, menük, űrlapmezők. Ezek az alap stílusukkal is alkalmazhatók, de a felhasználó személyre is szabhatja őket.

A használatuk egyszerű, ami megkönnyíti a fejlesztési folyamatot, és egységes kinézetet biztosít az alkalmazásnak.

## 3.1.2. AngularFire[[20]](#AngularFire)

Az AngularFire[[20]](#AngularFire) könyvtár a Firebase[[2]](#GoogleFirebase) adatbázissal való kapcsolatot valósítja meg. Az itt definiált metódusok használhatók többek között autentikációra, valamint adatbáziselemek lekérésére és manipulációjára.

## 3.2. Google Firebase[[2]](#GoogleFirebase)

A Firebase[[2]](#GoogleFirebase) egy Google által fejlesztett backend platform alkalmazások fejlesztéséhez. Széleskörű szolgáltatásokat nyújt a felhasználók számára, és különféle platformokon való alkalmazások fejlesztését támogatja.

## 3.2.1. Authentication

A Firebase Authentication szolgáltatás a felhasználók autentikációját könnyíti meg. Többféle bejelentkezési módot is lehetővé tesz, az e-mail címen kívül használható például telefonszám, Google és Facebook fiók is. Nemcsak felgyorsítja az alkalmazáson belüli autentikáció fejlesztését, de a felhasználói adatok biztonságos tárolásáról is gondoskodik.

## 3.2.2. Cloud Firestore

A Cloud Firestore adatbázis NoSQL dokumentumokként tárolja az adatokat. A lekérdezésekhez beépített metódusok használhatók, amik egyszerűbbé teszik az adatbázissal való kommunikációt.

## 3.2.3. Cloud Storage

A képek tárolásához a Cloud Storage szolgáltatást használtam, ami egy adatbázis fájloknak.

## 3.2.4. Hosting

Az alkalmazást a Hosting szolgáltatás segítségével hosztoltam. Ehhez a Firebase[[2]](#GoogleFirebase) által biztosított subdomaint is igénybe vettem.

## 3.3. Tesseract.js[[7]](#TesseractJs)

A Tesseract.js[[7]](#TesseractJs) egy JavaScript nyelvre épülő OCR motor. Több, mint 100, köztük magyar nyelven is képes szöveget felismerni, emellett az automatikus szövegelrendezést és a szkriptek detektálását is támogatja.

A kódba szkript tagként beimportálható, és ezek után egy workerként vagy egy időzített feladat elvégzésére is használható.

## 3.4. Sass[[5]](#SCSS)

A Sass[[5]](#SCSS) egy CSS bővítmény és előfeldolgozó, ami a CSS-hez hasonló, de hatékonyabban használható szintaktikával rendelkezik. A stíluslapokba új funkciókat vezet be, mint a változók, az egymásba ágyazott szabályok és az importálások.

Két szintaktikával rendelkezik, amik közül és az SCSS[[5]](#SCSS)-t használtam. A kettő közül az SCSS[[5]](#SCSS) jobban hasonlít a CSS-hez, mivel itt is ugyanúgy szükséges a kapcsos zárójelek és a pontosvesszők használata, míg a SASS[[5]](#SCSS) szintaxis esetében ezek szóközökkel és vesszővel helyettesítettek.

## 3.5. Egyebek

A diagramokat a draw.io[[3]](#drawio) oldal segítségével készítettem. Itt több szoftverfejlesztéshez kapcsolódó diagram sablonja elérhető, valamint előre elkészített elemeket is biztosít.

A fejelsztést a Visual Studio Code[[4]](#VisualStudioCode) szerkesztőben végeztem. Az alkalmazás átláthatóvá teszi a kódot, rendelkezik beépített terminállal, lehetőséget ad bővítmények hozzáadására, és rengeteg olyan funkcióval rendelkezik, ami felgyorsítja a fejlesztés folyamatát.

A GitLab[[6]](#GitLab) platformját egyaránt használtam verziókövetésre és a feladatok kezelésére is. Itt követtem nyomon a fejlesztés haladását, valamint a feladatok használatával a következő lépéseimet is előre el tudtam tervezni.

# Irodalomjegyzék

1. Angular: <https://angular.io/> (Utolsó megtekintés: 2024. 01. 22.)
2. Google Firebase: <https://firebase.google.com/> (Utolsó megtekintés: 2024. 01. 18.)
3. draw.io: <https://app.diagrams.net/> (Utolsó megtekintés: 2024. 01. 18.)
4. Visual Studio Code: <https://code.visualstudio.com/> (Utolsó megtekintés: 2024. 01. 18.)
5. Sass, SCSS: <https://sass-lang.com/> (Utolsó megtekintés: 2024. 01. 18.)
6. GitLab: <https://about.gitlab.com/> (Utolsó megtekintés: 2024. 01. 18.)
7. Tesseract.js: <https://tesseract.projectnaptha.com/> (Utolsó megtekintés: 2024. 01. 18.)
8. Zoho Expense: <https://www.zoho.com/expense/> (Utolsó megtekintés: 2024. 01. 21.)
9. Verify: <https://www.veryfi.com/> (Utolsó megtekintés: 2024. 01. 21.)
10. Expensify: <https://www.expensify.com/> (Utolsó megtekintés: 2024. 01. 21.)
11. Shoeboxed: <https://www.shoeboxed.com/> (Utolsó megtekintés: 2024. 01. 21.)
12. Dext Prepare: <https://dext.com/en/products/prepare> (Utolsó megtekintés: 2024. 01. 21.)
13. Invoice Maker, Invoice Calculator, Invoice Generator: <https://saldoinvoice.com/invoice-calculator/> (Utolsó megtekintés: 2024. 01. 21.)
14. EveryoneAllin: <https://everyoneallin.com/bill-splitter/online-bill-splitter/> (Utolsó megtekintés: 2024. 01. 21.)
15. Best Bill Splitter: <https://bestbillsplitter.com/> (Utolsó megtekintés: 2024. 01. 21.)
16. Kittysplit: <https://www.kittysplit.com/en/> (Utolsó megtekintés: 2024. 01. 21.)
17. IRIS KashFlow: <https://www.kashflow.com/> (Utolsó megtekintés: 2024. 01. 21.)
18. xero: <https://www.xero.com/> (Utolsó megtekintés: 2024. 01. 21.)
19. Angular Material: <https://material.angular.io/> (Utolsó megtekintés: 2024. 01. 22.)
20. AngularFire: <https://firebaseopensource.com/projects/firebase/angularfire/> (Utolsó megtekintés: 2024. 01. 22.)

# Nyilatkozat

Alulírott Somogyvári Hanna programtervező informatikus szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Szoftverfejlesztés Tanszékén készítettem, programtervező informatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Diplomamunka Repozitóriumában tárolja.

Dátum

Aláírás

# Köszönetnyilvánítás