# 1.Bevezető

A mai rohanó világban a bevásárlások egyre növekvő százaléka az interneten történik, mindez lehetőséget nyújtva a vásárlóknak, hogy egy bizonyos terméket több, akár hazai akár külföldi, oldalról is megvásárolhasson. Az e-commerce-el foglalkozó cégek rohamos fejlődésnek indultak az utóbbi évtizedben mely maga után vonja az érdekesebbnél érdekesebb marketing fogásokat, melyekkel a célközönséget próbálják vásárlásra bírni.

Valószínűleg mindenki hallott már a “Black Friday” az-az „Fekete Péntek” -nek nevezett jelenségről amely inspirációként szolgált az alkalmazás megvalósításához. Ez a kifejezés legelőszőr az 1800-as években fogalmazódott meg, amikor is Jay Gould és James Fisk az amerikai arany árak manipulálása által 20%-os esést okoztak a részvénypiacon melynek következtében az árucikkek értéke felére csökkent. A 20. század közepe fele ez már egészen más jelentéssel bírt, ugyanis a Hálaadás ünnepét követő napon, az-az pénteken vette kezdetét a karácsonyi árleszállítás, mely sok cég esetében életmentő volt, hiszen ekkor kerültek át a veszteséges állapotból melyet pirossal jelöltek, a jövedelmezőbe, amit már fekete írószerrel jegyeztek fel. Ebben az időszakban a megszokottnál jóval nagyobb és több árleszállítással vonzották az embereket.

Mint azt sokan tudjuk, országunkban is nagy népszerűségnek örvend ez a jelenség, habár eléggé távol áll az eredeti koncepciótól. Nagyon sok mesterséges árleszállítással próbálják becsapni az embert, melyet legtöbb esetben jól kitervelt ár ingadozással oldanak meg. Ugyanakkor, nem kizárólag ebben a periódusban lehet észrevenni az úgymond „hamis” kedvezményeket ezért szükségét láttuk egy olyan alkalmazás kifejlesztésének, amely nyomon tudja követni egy megadott termék árat, illetve annak ingadozását.

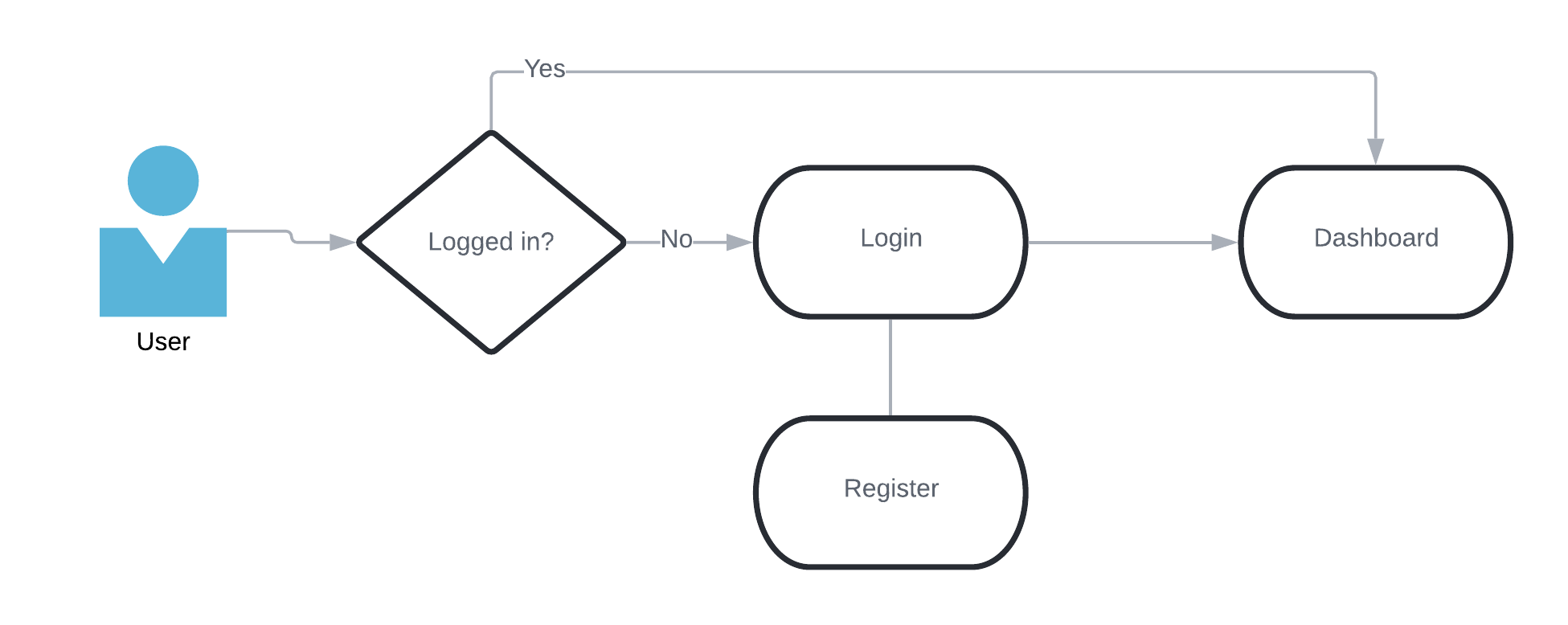
Mivel az interneten publikus adatok találhatók, ezek felhasználásával semmiéle kár nem keletkezik az adott weboldalak számára. Azt a HTML kódot, amelyből az információkat kinyerni szeretnénk, bármelyik felhasználó megtekintheti az F12 billentyű megnyomásával vagy az adott oldalon jobb click-et követően az “Inspect” opcióra kattintva. Az alkalmazás ezen nyilvános adatok feldolgozásával és elemzésével kivan foglalkozni.

# 2. Célkitűzések

* Az alkalmazás célja, hogy segítsen a vásárlónak reális képet alkotni egy termék áráról, illetve annak változásáról. Fontosnak tartottuk, hogy az alkalmazást egyszerűen és intuitívan lehessen kezelni, azaz bármilyen funkció pár érintésre legyen a felhasználótól
* Rövid használati útmutató, szóvéges formában
* Regisztrálási lehetőség
* Be jelentkezési lehetőség biztosítása, hogy különböző eszközökön is elérhetőek legyenek a követett termékek információi
* Felhasználói fiók jelszavának változtatási lehetősége
* Felhasználói fiókból való ki jelentkezés
* Felhasználói fiók törlésé
* A termékek ábrázolása lista szerűén történjen
* Az árak változását a felhasználó számára vizuális formában szeretnénk ábrázolni a könnyebb átláthatóság érdekében
* Az árak időben való változását könnyen átlátható vonal diagramon ábrázolni
* Egyszerű átirányítás a termék oldalára
* Támogatott oldalak listázása
* Telefonos alkalmazás
* Kiegészítő, Chrome, illetve Firefox alapú böngészőkre

# 3. Felhasználói Követelmények

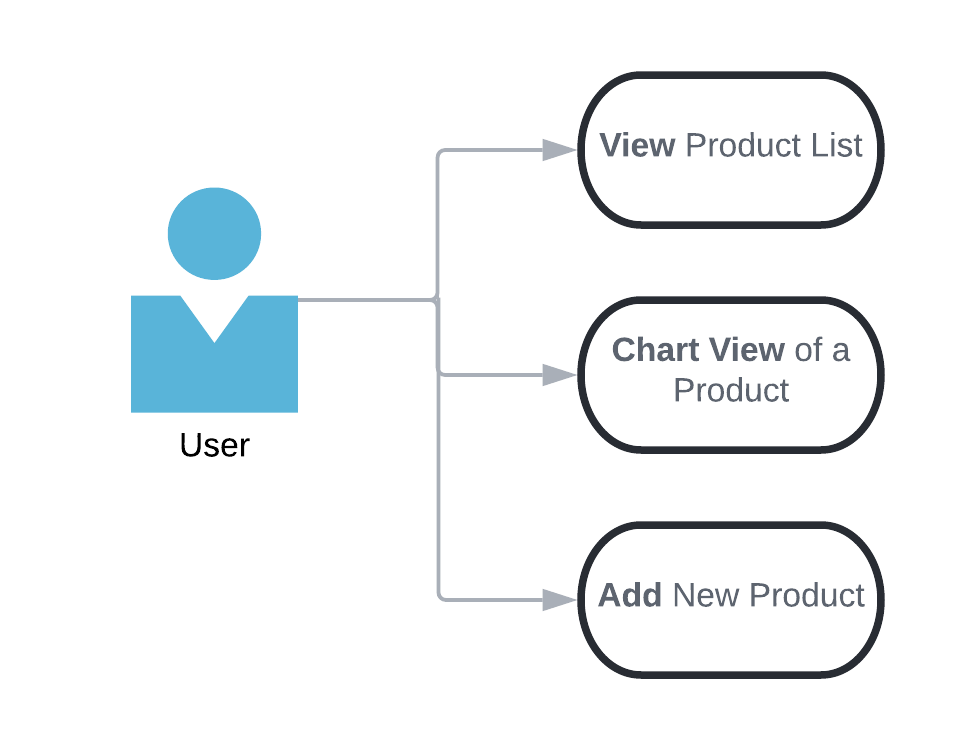
A felhasználónak mindenekelőtt be kell tudnia jelentkezni, ahhoz, hogy elérje a terméklistáját, ez azért fontos, mivel eszköz váltás esetén nem szeretnénk elveszíteni az addig követett termékeket. A bejelentkezéshez szükséges adatok a regisztrációkor megadott email cím, illetve jelszó. Mindezt az 1.ábra illusztrálja. A felhasználó jelszava base64 hash általi titkosítással van eltárolva, ezáltal az nem elérhető eredeti formájában, ezt a funkciót a Firebase Authentication valósítja meg, amely jelszó módosítási lehetőséget is lehetővé tesz.



( 1.ábra Login Activity )

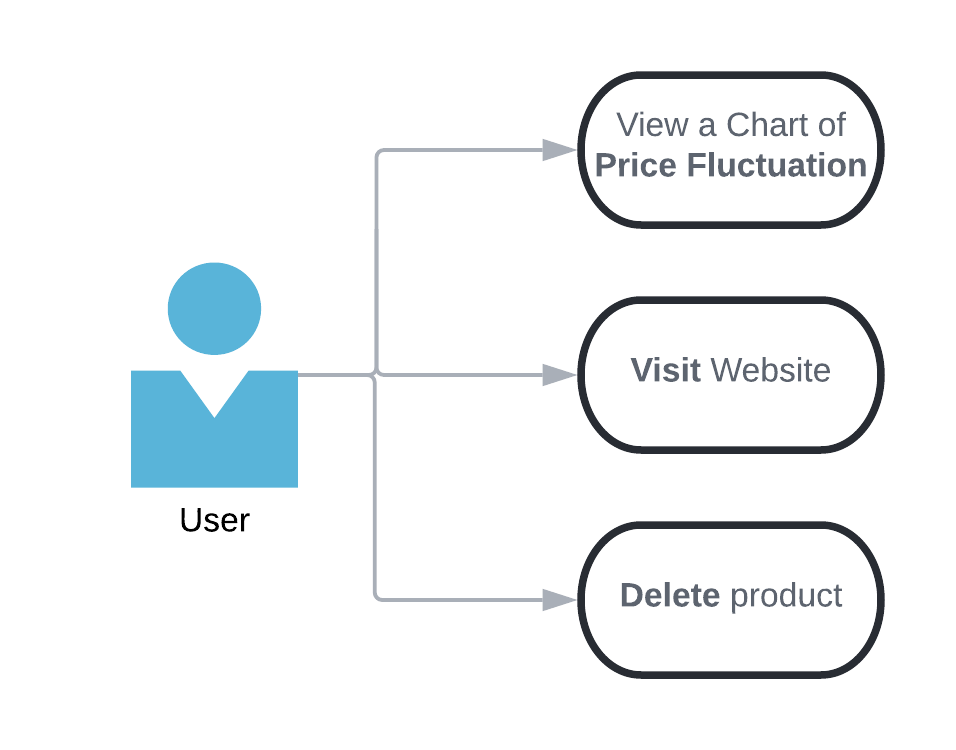
Abban az esetben, ha a felhasználó nincs regisztrálva, megteheti ezt a „ Not registered? Click here! ” szövegre kattintva. A regisztrációhoz szükséges egy érvényes e-mail cím, valamint jelszó megadása. Sikeres regisztrálás esetén a felhasználó vissza kerül a bejelentkező ablakra, ahol be tud lépni, azzal a feltétellel, hogy az automatikusan küldött levél által visszaigazolta email címét.

A bejelentkezést követően a felhasználó a főoldalra kerül, ahol a követett termékek listáját tekintheti meg. A listában minden egyes elemnek látható a megnevezése, egy a terméket ábrázoló kép, illetve az adott termék aktuális ára. Továbbá ezen az oldalon lehetősége van a felhasználónak új termékeket hozzá adni a listához, lásd 2.ábra.

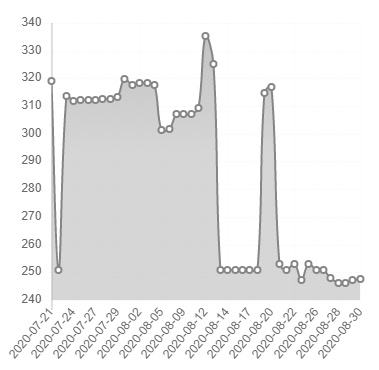


( 2.ábra Dashboard View )

Egy a listában lévő elemre kattintva, az alkalmazás átvisz egy másik oldalra, ahol több információt kapunk a követett termékről. A termék árát tartalmazó gombra kattintva, egy grafikonon tekinthetjük meg a termék árának változását a hozzáadás napjától az aktuális dátumig, lásd 3.1 ábra. Az „ See product page ” gombra kattintva az alkalmazás megnyitja a terméket tartalmazó weboldalt. Ugyanitt található a törlés gomb, melyre kattintva a termek törlésre kerül a listából és nem fogjuk tovább követni, lásd 3.ábra.

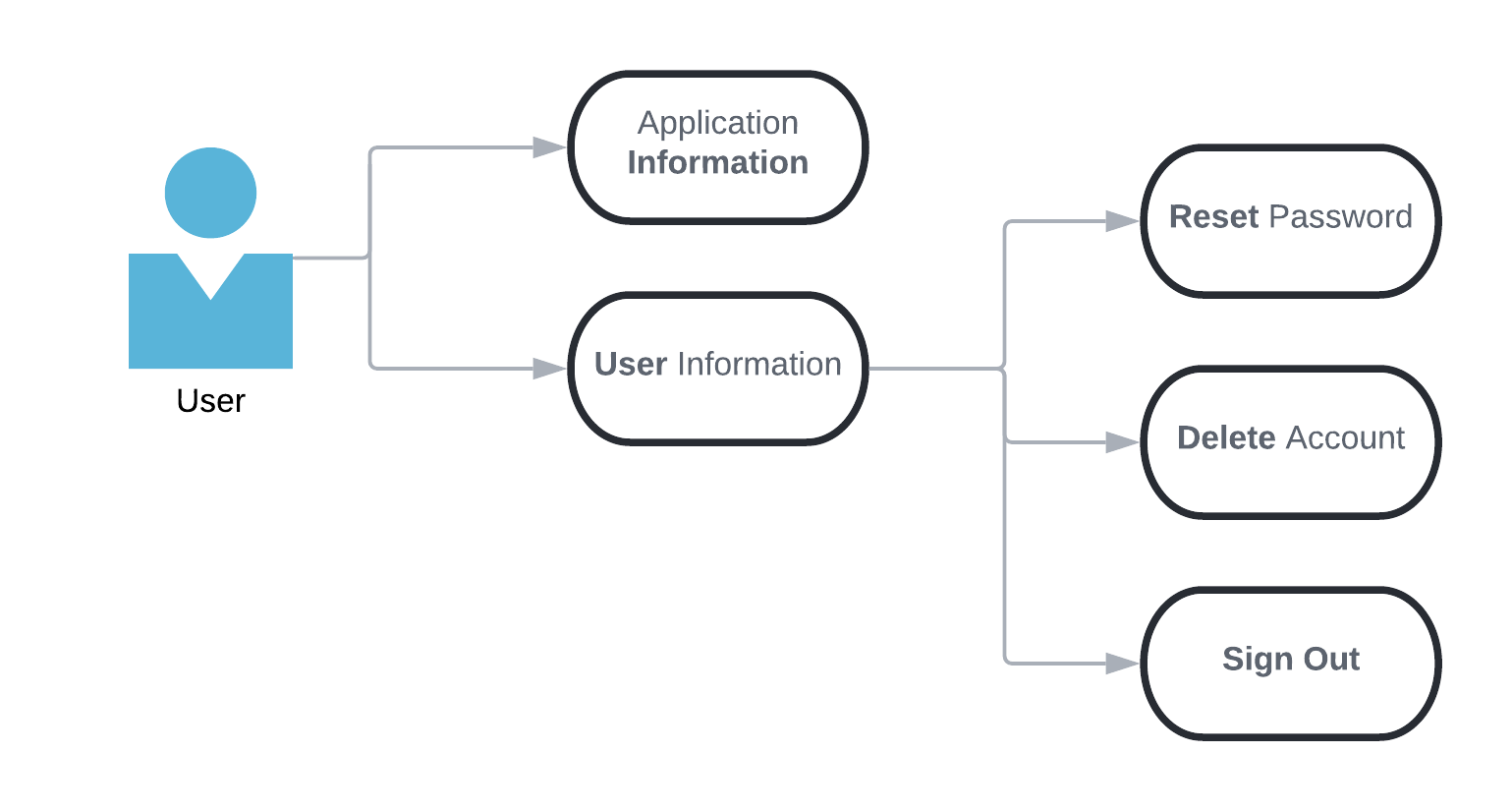


( 3.ábra Chart View )



( 3.1 ábra Chart of Price Change )

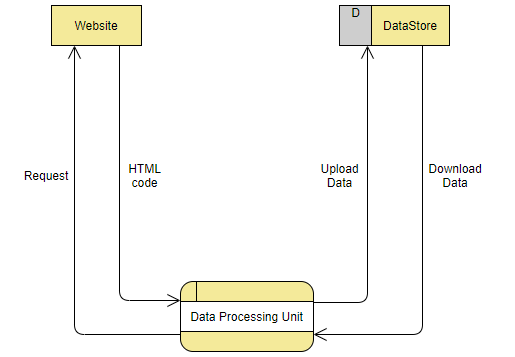
A felhasználó rendelkezésére áll továbbá két menü. Egy információs, amely röviden leírja az alkalmazás használatát, illetve tartalmazza az általa támogatott weboldalak listáját. A másik a felhasználó fiókjával kapcsolatos információkat és funkciókat tartalmaz. Itt tekintheti meg a felhasználó, hogy milyen email címmel jelentkezett be, megváltoztathatja az aktuális jelszavát, törölheti a fiókját, illetve kijelentkezhet az alkalmazásból, lásd 4.ábra.



( 4.ábra Additional Information )

# 4. A rendszer leírása

## 4.1 Backend



( Data Flow Diagram created by Visual Paradigm Online )

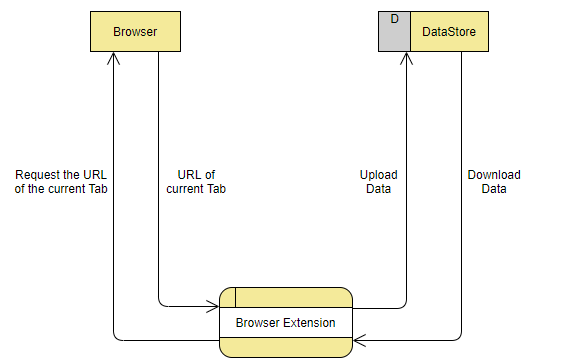
Ahhoz, hogy a szoftver működőképes legyen szükséges egy rendszer, amelyen futtatni tudjuk. A rendszernek képesnek kell lennie Python 3-as kódot futtatnia, fel kell legyen telepítve a Python 3.8.0, valamint szükséges egy stabil, azaz mindig aktív internet kapcsolat (minimális fel-le töltési sebesség ne essen 10 Mb/s alá) mivel a rendszer ezen keresztül kéri le, illetve tölti fel az információkat az adatbázisba.

A Python kód futtatásához, szükséges néhány, a beépített függvény könyvtárat kiegészítő, specifikus könyvtár. Ezek nélkül a program nem fog helyesen lefutni, ezért a következő elemeket kell hozzáadni: bs4, dacite, firebase, gcloud, requests, schedule, time. A requests könyvtár segítségével tudunk HTTP kéréseket intézni, amely ahhoz szükséges, hogy megkapjuk egy adott termék linkjéhez tartozó HTML struktúrát. A visszakapott nyers kódban nehezen lehetne kereséseket végrehajtani, ezért ezt parse-oljuk a bs4 által megvalósított BeautifulSoup nevű függvénnyel, amely egy kereshető struktúrát hoz létre több hasznos funkcióval, a neki megadott HTML kódból. Mikor mindez megvan és feldolgoztuk a szükséges adatokat, a firebase illetve gcloud könyvtárak segítségével tudjuk elérni a használt adatbázist, melyre feltöltjük az adatokat a megfelelő formában. Adatok lekérésé során szükséges a dacite könyvtár által megvalósított dictionary parser, amely az adatbázis által visszatérített dict struktúrából, egy általunk megvalósított osztályt felhasználva hoz létre, egy, az osztályhoz tartozó példányt, mellyel ezek után dolgozni tudunk. A schedule illetve time könyvtárak az árak illetve termékek időszakos ellenőrzésére szükségesek.

Az adatfeldolgozás megkönnyítésé érdekében, több saját osztály is szükséges volt létrehozni. Ezek közül fontosabbak a ProductData, amely egy termekről tárolt információkat tartalmazza, NewData, amely egy új link hozzáadásakor használandó, illetve UtilProduct, mely a fentebb említett parsolaskor fontos.

## 4.2 Frontend

### 4.2.1 Böngésző kiegészítő



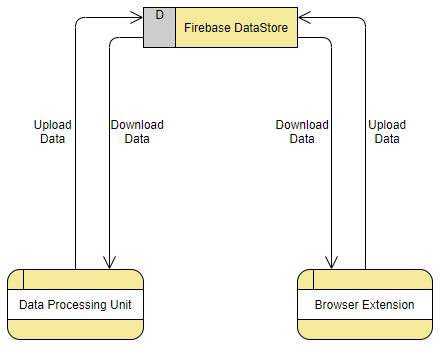
( Data Flow Diagram created by Visual Paradigm Online )

A kiegészítő használatához szükséges egy kompatibilis böngésző, ami chromium alapú. Egy rövid regisztrálás majd bejelentkezés után használható is az alkalmazás. A felhasználó információi, legyenek azok a fiókjához vagy a termék listájához tartozóak, az adatbázisban tárolódnak el, ezért létfontosságú, hogy a megfelelő internetkapcsolat álljon rendelkezésünkre.

A felhasználói felület elkészítéséhez is felhasználásra került pár külső könyvtár: firebase-app, firebase-auth, sweetalert2, fontawesome, bootstrap. Adatbázis kapcsolatra a firebase-app könyvtárat kellett használni, amelynek egyszerűen az adatbázis elérhetőségét kellett megadni, onnantól a létrejött példányon keresztül lehetett végezni a műveleteket. A firebase-auth könyvtár tette lehetővé, hogy gyorsan és egyszerűen lehessen megoldani a regisztrálást, illetve bejelentkezést. Regisztrálást követően a felhasználó egy megerősítő email-t kap, ahol a visszaigazolás után, aktívvá válik a fiókja és be tud jelentkezni. Fontos, hogy amíg a megerősítés nem történt meg, addig a felhasználónak nincs lehetőségé bejelentkezni. A felület vizuális felülete bootstrap, fontawesome, valamint sweetalert2 elemek segítségével került megvalósításra, ahol a bootstrap a fő struktúrákért, míg a sweetalert, a beugró ablakokért felelős. A fontawesome az ikonok vagy betűtípusok formázásában kapott szerepet.

### 4.2.2 Telefonos alkalmazás

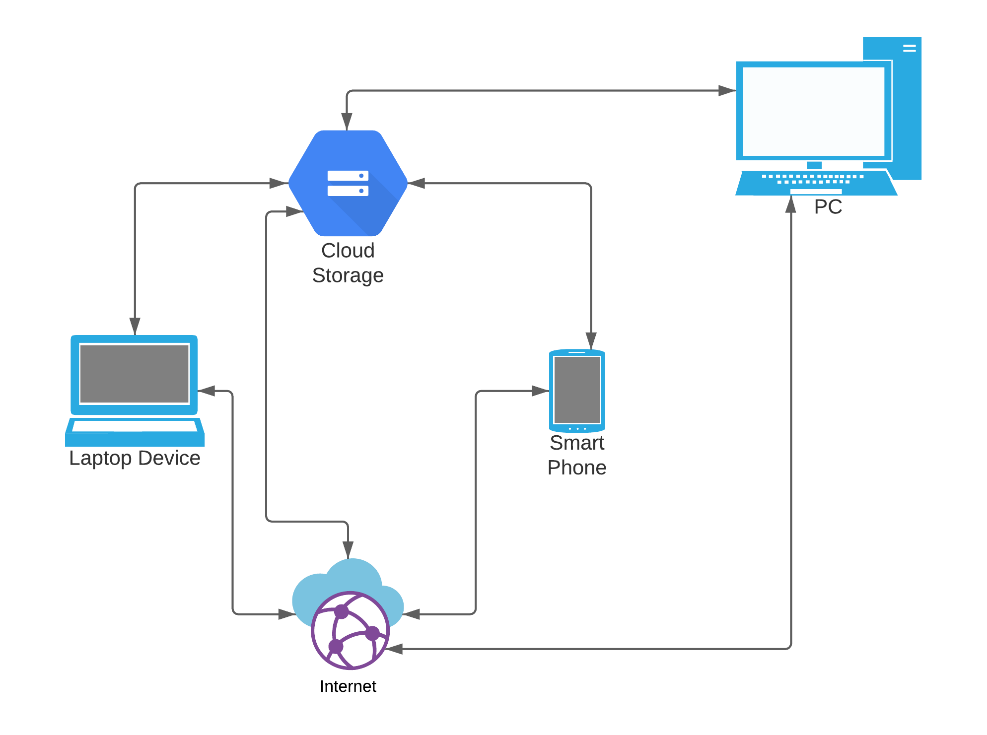
## 4.3 Adatbázis



( Data Flow Diagram created by Visual Paradigm Online )

A FireBase mobil és webes applikációt fejlesztő platformot választottuk adatbázisunk létrehozása érdekében, ugyanis ez kitűnőnek minősült célunk elérésére. Segítségünkre szolgált az, hogy ez nem egy megszokott adattáblákkal rendelkező adatbázis, hanem egy nosql, valós idejű rendszer, mely lehetővé teszi a felhasználó számára a pillanatnyi adatok láthatóságát, amit annak köszönhetünk, hogy az adat fel és letöltése valós időben történik.

# 5. Architektúra



( 5. ábra A rendszer architektúrája )

A rendszer alapvető részét képezi az adatbázis, amely az adatok tarolását illetve az azokat elérő API-t szolgáltatja. Ehhez az adatbázishoz két alapvető típusú készülék csatlakozik. A felhasználó oldali, amely lehet akár böngésző kiegészítő vagy telefonos alkalmazás, illetve a szerver vagy logika oldali, amely az adatok feldolgozását biztosító logikát és erőforrást tartalmazza, utóbbit a fent látható 5. ábrán a PC-vel jelölt eszköz jelképezi.

A böngésző kiegészítő szükséges, mivel sokkal gyorsabban el lehet érni a megtekinteni kívánt adatokat, valamint sokkal egyszerűbbé teszi a termékek hozzáadását azzal, hogy a követni kívánt termék oldalát egyáltalán nem kell elhagyni. Továbbá fontos, hogy az alkalmazás tudja azt, hogy a felhasználó épp milyen oldalon tartózkodik, ahhoz, hogy a megfelelő URL-t kapja meg, mindezt a kiegészítő könnyen és megbízhatóan el tudja végezni.

Mivel azonban nem mindig vagyunk laptop vagy asztali gép közelben, ezért lehetőséget nyújtunk arra, hogy telefonos applikáció segítségével is el lehessen érni a követett terméket, illetve újat lehessen hozzáadni a követő listához. Az alkalmazásunk ezen mobilos része még nem került kivitelezésre.

Ahhoz, hogy az eszközök kommunikálni tudjanak egymással, egyértelmű, hogy szükség van valamiféle összeköttetésre, amely a mi esetünkben az internet lesz. Ez a funkció létfontosságú, mivel minden adatot fel illetve le kell tölteni az adatbázisból, függetlenül az eszközök tartózkodási helyétől.