Marvel Szuperhősök

Szoftver Rendszerek Tervezése

Barabási Barna, Izsák Andrea, Kakucs Botond
Sapientia EMTE, Marosvásárhelyi Kar

Tartalomjegyzék

Bevezető

Napjainkban az okostelefonok szinte elengedhetetlenné váltak. Ezek az eszközök napunk számos tevékenységében besegítenek, valamint megkönnyítik egyes feladatainkat.

Az okostelefonok által kínált szórakozási lehetőségek listája terjedelmes, többféle kategória közül választhatunk. Sokszor találkozunk ismertető célú alkalmazásokkal, hiszen nagy cégek is foglalkoznak, ezen alkalmazások fejlesztésével. Amennyiben kikapcsolódási időnkben szeretnénk többet megtudni a Marvel univerzum szuperhőseiről, az általunk fejlesztett alkalmazás negy jó kiindulópont lehet.

Az alkalmazás a Marvel általi kínálat API(application programming interface) segítségével hozzáfér egy adatbázishoz, amely ezen univerzum hőseiről tárol információkat, képeket.

Rendszerkövetelmények

Nem funkcionális rendszerkövetelmények:

- Android 6.0+ operációs rendszer
- Minimális API szint 23
- Minimális memóriaigény 1 MB
- Minimális tárhely 8 MB

Funkcionális rendszerkövetelmények:

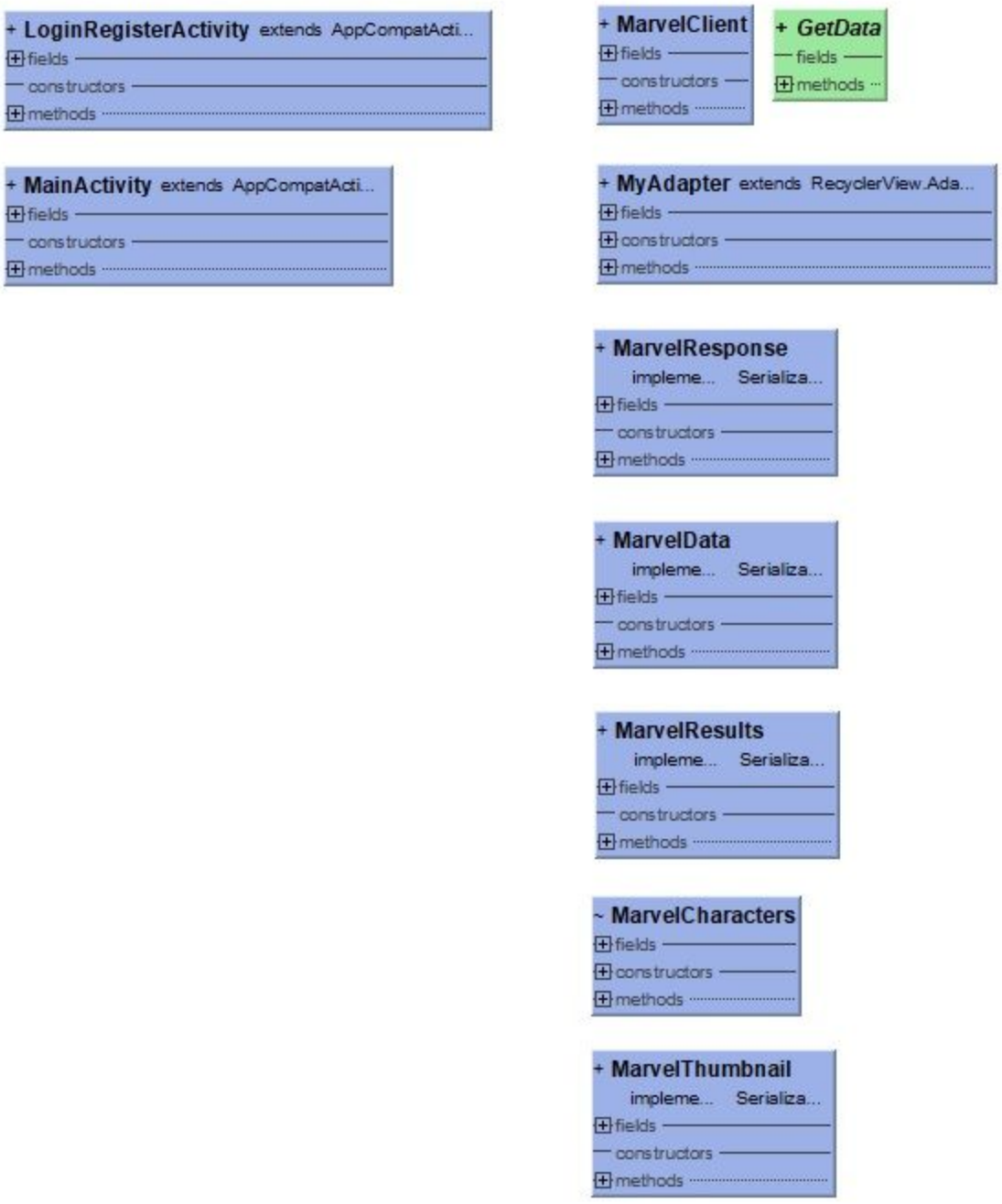
- Fejlesztői környezet: Android Studio 3.0
- Funkciók implementálása Java-ban történt
- Engedély a internet használatára

- Adatbázissal való kommunikáció

Felhasználói követelmények

- Android operációs rendszert futtató okostelefon
- Menüpont ahol regisztrálni lehet
- Menüpont ahol bejelentkezni lehet
- Felület ahol véletlenszerű hősök, valamint keresési felület jelenik meg
- Internetkapcsolat

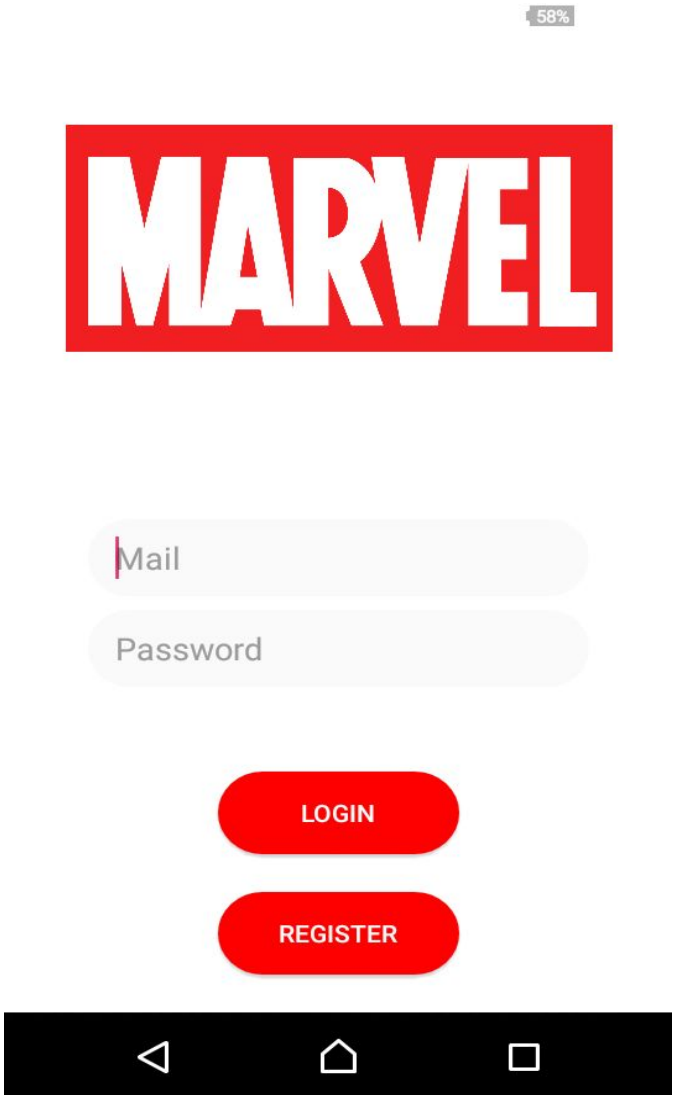
Diagrammok



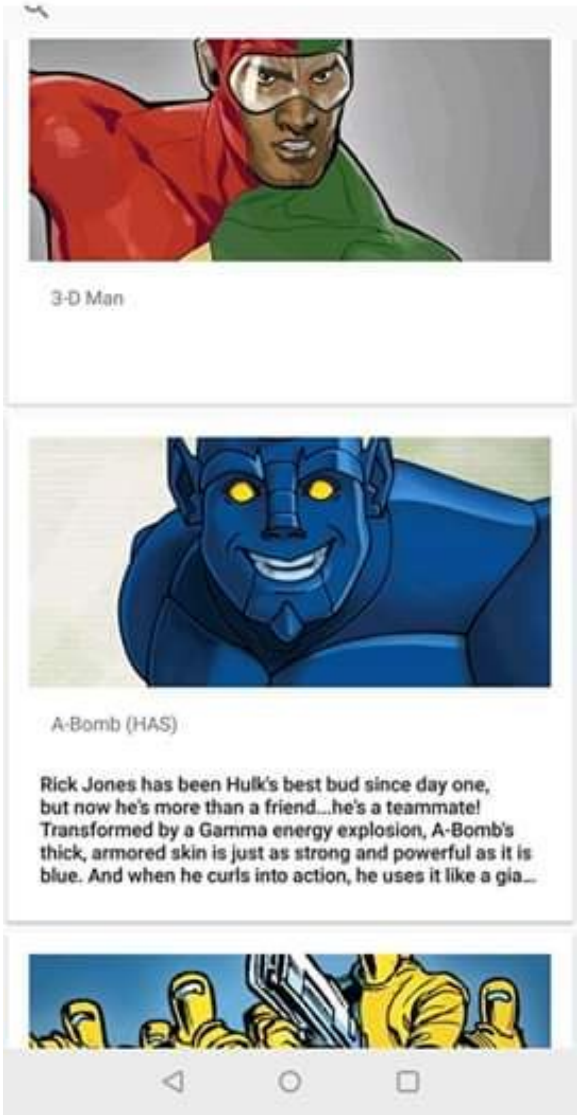
A fenti diagramon látható az általunk készített alkalmazás szerkezete, valamint azok a folyamatok, amelyek a háttérben történnek meg és biztosítják a felhasználó számára az alkalmazás akadálymentes használatát. Két külső erőforrással történik kommunikáció, a Firebase-el, amely tárolja az általunk feltöltött adatokat, valamint a Marvel Api, amely szolgáltatja azokat az adatokat, amelyek megtalálhatóak a Marvel cég adatbázisában és publikusak mindenki számára.

A szoftver használata

Az alkalmazás elinditását követően a felhasználó a bejelentkezési és a regisztrálási képernyővel szembesül, ahol beléphet, mint már létező felhasználó az adatbázisban, illetve ha még nem használta korábban az alkalmazást létrehozhat egy új felhasználót mindössze az email címe és egy jelszó megadásával.



Sikeres regisztráció esetén a bejelentkezési képernyőt felváltja a listázási képernyő, amelyen megtekinthetőek a kívánt adatok amelyeket a Marvel API biztosít a felhasználó számára. Az általunk keresett adat gyorsabb megtalálásában egy keresősáv áll rendelkezésünkre.



A keresett tartalom megtalálása után lehetőségünk van elmenteni azt úgymond a kedvencek közé, ezzel elősegítve az alkalmazás gyorsabb és egyszerűbb használatát.

Android - Kódrészletek

```
GetData service = MarvelClient.getRetrofitInstance().create(GetData.class);

Call<MarvelResponse> call = service.getAllCharacters(Long.valueOf(ts), API_KEY, HASH);
//Call<MarvelResponseTest> call = service.getCharacterByName("Iron%20Man",Long.valueOf(ts), API_KEY, HASH);
```

Létrehozunk egy handler-t a Retrofit interface-nek, valamint egy MarvelResponse tÍpusú hivást inditunk el, amely a GetData servicenek továbbadja a jelenlegi időbélyeget, a privát API kulcsunkat , valamint egy HASH kulcsot, amely egy md5 feldolgozása a jelenlegi időbélyeg, publikus kulcs és privát kulcsnak.

```
public void onBindViewHolder(CustomViewHolder holder, int position) {

        holder.textCharacter.setText(dataList.get(position).getName());
        holder.textDescription.setText(dataList.get(position).getDescription());
    Uri uri = Uri.parse(dataList.get(position).getThumbnail());
    Glide.with(holder.imageView.getContext()).load(uri).into(holder.imageView);
    Log.d(TAG,  msg: "onBindViewHolder() called with: holder = [" + uri + "], position = [" + position + "]");


}
```

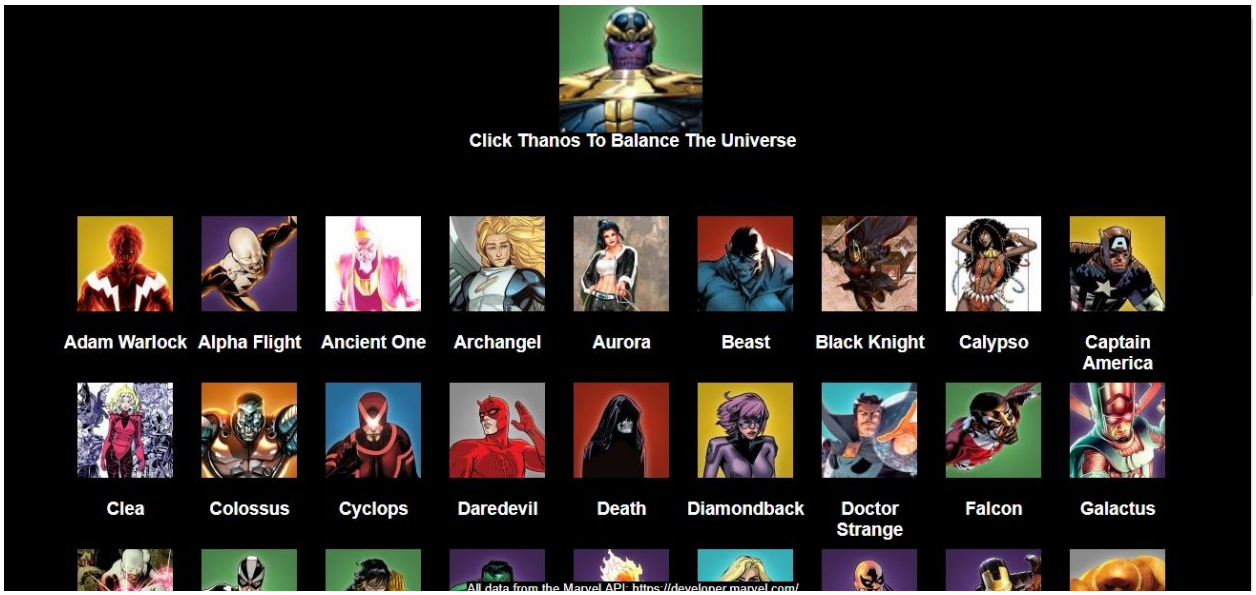
A holder felépÍtése: hozzárendeljuk a holder megfelelő elemeihez, a megfelelő karakter adatait. A karakter képét Glide segitségével sikerult megvalósÍtani.

Webes felület - Kódrészletek

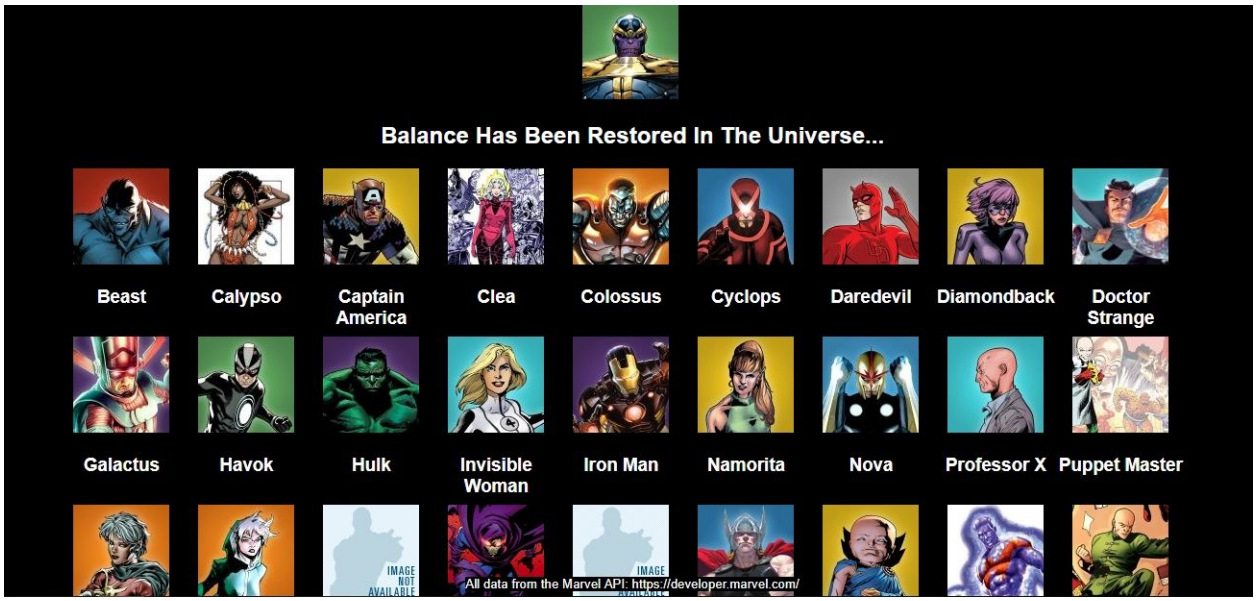
A Webes felület célja, Thanos híressé vált csettintésének illusztrálása.

A Végtelen háború végén, mint ismeretes, Thanos elérvén célját egy laza csettintéssel kiirtotta a univerzum fél lakosságát, köztük jó néhány szuperhőst és egyéb harcos csatlakozót, így elveszítettünk jó néhányat a Marvel moziuniverzumos kedvenceink közül.

A felületen a hősök vannak feltüntetve, melyek Thanos fényképére kattintása után, egy temetés zenei aláfestés keretén belül, a karakterek sorra elhalnak.



Miután az egyensúly vissza lett állítva az univerzumban:



Visual Studio Code környezetet használtunk a programozáshoz.

Amikor a VS Code készült, figyelembe vették a fejlesztői igényeket. Mivel minden fejlesztő máshogy kódol, talán az okos univerzalitás lett a megoldás. Open source, ami dicséretes.

- nem zabálja a memóriát
- nem indexel be minden megnyitott fájlt feleslegesen
- nincs tele olyan funkciókkal, amiket biztos, hogy sohasem fogok használni.
- ugyanakkor eszméletlen sok kiegészítő készül hozzá, érthető és könnyű bővíthetőségéből adódóan.

A webesprojekt három részből tevődik össze, egy javascript, egy html és egy css.

JavaScript:

A JavaScript és a Java nyelvnek egyaránt vannak előnyei és hátrányai eseménykezelők írásakor. Az olyan fejlesztők számára, akiknek csak az egyik nyelvben járatosak, az

ismerős nyelv használatának előnye nyilvánvaló, mások számára azonban a döntés a jelentés követelményeitől függ.

- Könnyű hozzáadni egy egyszerű parancsfájlt egy bizonyos eseménykezelő számára
- Egyszerűbb nyelvi szerkezetek, szabadabb írás, kevésbé szigorú nyelvi szabályok
- Az eseménykezelők könnyen megkereshetők és megjeleníthetők
- Elérhető egy integrált hibakereső

javaScript segítségével történik meg a projekt kódolásának nagy része, itt vannak leimplementálva a függvények, amelyek lekérik az adatokat a Marvel API-tól és elvégzik a szükséges műveleteket

```
function kill(characters, leftToDie) {
  if (leftToDie > 0) {
    const randomIndex = Math.floor(Math.random() * characters.length);
    const [characterChosen] = characters.splice(randomIndex, 1);

    characterChosen.style.opacity = '0.2';
    characterChosen.classList.remove('alive');
    characterChosen.classList.add('dead');

    console.log('Killing...', characterChosen.querySelector('h3').textContent);

    setTimeout(() => {
      characterChosen.style.transform = 'scale(0)';
      characterChosen.style.width = '0px';
      characterChosen.style.height = '0px';
      kill(characters, leftToDie - 1);
    }, 1300);
  } else {
    theTruth.style.opacity = '1';
    fadeOutFuneralMusic();
  }
}
```

HTML:

A nyelv lényege: Egy megírt szövegbe olyan jelöléseket helyezünk, amelyet az ún. HTML értelmezôk feldolgoznak, és ezáltal kialakulnak a szöveg-szöveg, szöveg-kép stb. kapcsolatok.

A megírt szövegek ASCII formátumúak, a HTML parancsok 'kisebb'-'nagyobb' jelek között szerepelnek, bármely szövegszerkesztôvel írhatóak.

A nyelv elônye, hogy nincs "kódtábla-probléma"

html-ben valósítottuk meg a háttérzenét:

```
<audio id="intro-sound" src="sounds/intro.mp3"></audio>
<audio id="snap-sound" src="sounds/snap.mp3"></audio>
<audio id="funeral-sound" src="sounds/funeral.mp3"></audio>
```

CSS:

A CSS a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezen kívül használható bármilyen XML alapú dokumentum stílusának leírására is, mint például az SVG, XUL stb.

Projektünkben a **css** arra szolgált, hogy a felületet formáztuk (a karakterek elrendezését és megjelenítését is ezzel valósítottuk meg)

```
.characters {
  display: flex;
  flex-wrap: wrap;
  width: 90%;
  z-index: -1;
  transition-duration: 1s;
}

#thanos {
  margin-top: 10px;
  z-index: 100;
}

#thanos.hover:hover {
  transform: scale(1.5);
}

#snap {
  height: 80%;
  width: auto;
  transition-duration: 4s;
  position: absolute;
  margin: 0 auto;
  top: 120px;
```

Jövőbeli tervek

Az az adatok feldolgozásának megvalósÍtása a Marvel API segÍtségével túl sok időt vett fel, Így a Searchbar implementálása jövőbeli terveink közé tartozik.

Felhasználó profiljának szerkesztése valamint kedvenc hősök elmentése is megvalósÍtási terveinkhez tartozik.

Bibliográfia

https://developer.android.com/

https://stackoverflow.com/

https://www.androidauthority.com/