

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Automatizálási és Alkalmazott Informatikai Tanszék

Nagy Viktor

**Kártyajáték megvalósítása** ASP.NET Core és Angular platformon

Konzulens

Dr. Kővári Bence András

BUDAPEST, 2021

Tartalomjegyzék

[Összefoglaló 4](#_Toc88088500)

[Abstract 5](#_Toc88088501)

[1 Bevezetés 6](#_Toc88088502)

[1.1 Általános bevezető 6](#_Toc88088503)

[1.2 Társasjátékok, internetes játékok 6](#_Toc88088504)

[1.3 Közös munka 6](#_Toc88088505)

[1.4 Saját feladatrészek 6](#_Toc88088506)

[2 Technológiák 7](#_Toc88088507)

[3 Tervezés, architektúra 8](#_Toc88088508)

[3.1 Feladat specifikáció leírása 8](#_Toc88088509)

[3.2 A rendszer architektúrájának felépítése 8](#_Toc88088510)

[3.2.1 Adatbázis 8](#_Toc88088511)

[3.2.2 Architektúra 8](#_Toc88088512)

[3.2.3 Use-case 8](#_Toc88088513)

[4 Program részletes bemutatása 9](#_Toc88088514)

[4.1 Backend 9](#_Toc88088515)

[4.1.1 Mikroszolgáltatások 9](#_Toc88088516)

[4.1.2 Többrétegű architektúra 9](#_Toc88088517)

[4.1.3 Adatbázis 9](#_Toc88088518)

[4.1.4 Adatelérési réteg 9](#_Toc88088519)

[4.1.5 Üzleti logikai réteg 9](#_Toc88088520)

[4.1.6 Prezentációs réteg 10](#_Toc88088521)

[4.1.7 API átjáró 11](#_Toc88088522)

[4.2 Frontend 11](#_Toc88088523)

[4.2.1 Komponsek 11](#_Toc88088524)

[4.2.2 Szervízek 11](#_Toc88088525)

[4.2.3 Pipe 11](#_Toc88088526)

[4.2.4 Guard 11](#_Toc88088527)

[4.2.5 Interceptor 11](#_Toc88088528)

[4.2.6 Snackbar 11](#_Toc88088529)

[4.2.7 WebSocker (SignalR) 11](#_Toc88088530)

[5 Összegzés, értékelés 12](#_Toc88088531)

[5.1 Továbbfejlesztési lehetőségek 12](#_Toc88088532)

[Irodalomjegyzék 13](#_Toc88088533)

[Függelék 14](#_Toc88088534)

Hallgatói nyilatkozat

Alulírott **Nagy Viktor**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2021. 11. 18.

...…………………………………………….

Nagy Viktor

Összefoglaló

Kedvcsinálás az elolvasásra, eredmények leírása.

Abstract

Ide jön a ½-1 oldalas angol nyelvű összefoglaló, amelynek szövege a Diplomaterv Portálra külön is feltöltésre kerül.

# Bevezetés

## Általános bevezető

Ma már szinte elképzelhetetlen lenne az élet informatika nélkül. Ez az, ami meghatározza az életünk rengeteg aspektusát. A mindennapjaink részévé váltak nem csak szórakozás téren, de akár megélhetési szinten is, ugyanis ma már ritka egy olyan munkahely, ahol ne használnának minimális szinten se informatikai eszközöket. Akár említhetem a szakdolgozat írását is, mert annak is konkrét követelményi vannak, hogy az adott szövegszerkesztőben mit, hogyan kell beállítani.

Ahogy halad az idő előre, úgy fejlődnek az

## Társasjátékok, internetes játékok

## Közös munka

## Saját feladatrészek

# Technológiák

# Tervezés, architektúra

## Feladat specifikáció leírása

## A rendszer architektúrájának felépítése

### Adatbázis

### Architektúra

### Use-case

# Program részletes bemutatása

Leírni, hogy a bemutatás az úgy fog felépülni, hogy a backend aljától halad a frontending

## Backend

Beszélni a .net és asp.netről

### Mikroszolgáltatások

A dockerről részletesen írni, miért jó használni, mi az előnye, milyen szinten használja a programunk, docker-compose. A mikroszolgáltatásokról írni, hasonlóan milyen előnyökkel jár. Képet bevágni a docker compose egyik conténeréről.

### Többrétegű architektúra

Nagyjából elővezetni, hogy milyen szinten lesznek kb szétosztva a fejezeteim, ugyanis a DAL, BLL, API rétegek mentén lesznek a nagyobb pontok. A rétegekről átfogóan beszélni, melyiknek mi a szerepe.

### Adatbázis

Elsőként az sql alapú adatbázisok előnyeiről/hátranyairól beszélni a nosql esekkel szemben, miért választottuk ezt. Majd a saját db-m részleteiről képek, felépítéséről.

### Adatelérési réteg

Elmondani, hogy ez az első szintje a 3 rétegnek, elmondani, hogy itt mi található.

#### Entity Framework

#### Code-First

#### DBContext

Kép

### Üzleti logikai réteg

Elmondani, hogy ez az második szintje a 3 rétegnek, elmondani, hogy itt mi található.

#### Command Query Responsibility Segregation minta

Beszélni a CQRS ről (mindegyiknél figyleni hogy elsőnek kiírni a teljes nevét), miért hasznos szétbontani commandokra és querykre a feladatokat.

#### Repository minta

Repositoryként működő storokról, ami egy plusz réteget ad, az egésznek.

#### Mediator minta

Mediatorról részletesen, ez szolgál a storeok és az api közötti résznek Itt zajlik a logika amiben nincs adatbázis művelet. Kép egy handle függvényről

#### Automapper

Szokásos bullshit, egyikről képet amiben van értelmes dolog is. (pl friend)

#### Globális kivételleképzés

Csak röviden a hellangról, egy saját kivételről kép is lehet.

#### Fontosabb logikai megoldások

Pl a friendek kezeléséről, lobby kezeléséről, inviteok kezeléséről lehet írni.

### Prezentációs réteg

Figyelni az API szó teljes kiírására elsőnek. Elmondani, hogy ez az harmadik szintje a 3 rétegnek, elmondani, hogy itt mi található.

#### Globális kivételkezelés (Hellang)

Csak kép a kódról a startupban és beszélni róla.

#### WebSocket technológia (SignalR)

Elmondani mire jó, majd megmutatni pár sajátot.

#### Authentikáció (IdentityServer4)

Az authentikáció flowjáról, tokenekről, konfigurációjáról.

#### Kontrollerek

Példa a mediator hívás indítására és elmondani, hogy a Commandok és Queryk micsodák.

#### Startup

Az elrendezésről is.

### API átjáró

Beszélni az ocelotról.

## Frontend

Beszélni az angularról, miért jó, miért ezt választottam hasonló.

### Komponsek

### Szervízek

### Pipe

### Guard

### Interceptor

### Snackbar

### WebSocker (SignalR)

# Összegzés, értékelés

Leírni egy általános összegzést, mi okzott nehézséget, mi ment könnyen, hogyan lehetett összehangolni a csapatmunkát, mennyire volt effektív. Mi alapján osztottuk szét.

## Továbbfejlesztési lehetőségek

Irodalomjegyzék

1. Levendovszky, J., Jereb, L., Elek, Zs., Vesztergombi, Gy.: Adaptive statistical algorithms in network reliability analysis, Performance Evaluation - Elsevier, Vol. 48, 2002, pp. 225-236
2. National Istruments: LabVIEW grafikus fejlesztői környezet leírása, <http://www.ni.com/> (2010. nov.)
3. Fowler, M.: UML Distilled, 3rd edition, ISBN 0-321-19368-7, Addison-Wesley, 2004
4. Wikipedia: Evaluation strategy, <http://en.wikipedia.org/wiki/Evaluation_strategy> (revision 18:11, 31 July 2012)

Függelék