

# Minecraft Router

## Projektfeladat specifikáció



**Informatikai Biztonsági és  
Adatvédelmi Tanácsadó Kft.**

# 1 Tartalomjegyzék

1	Tartalomjegyzék.....	2
2	Bevezetés .....	3
2.1	A feladat címe .....	3
2.2	A feladat rövid ismertetése .....	3
3	Elvárások a feladattal kapcsolatban.....	4
3.1	Operációs rendszer, környezet.....	4
3.2	Felhasználható programozási nyelv.....	4
3.3	Megoldás formátuma.....	4
3.4	Szoftverfejlesztés .....	4
3.5	Modulok.....	5
4	Specifikáció .....	6
4.1	Megjelenés.....	6
4.2	Funkciók.....	6
4.3	Proxy.....	6
5	Dokumentáció .....	7
5.1	Erőforrásterv, munkaidő-nyilvántartás.....	7
5.2	Üzemeltetői dokumentáció .....	7
5.3	Forráskód dokumentáció.....	7
5.4	Felhasználói dokumentáció .....	7
6	A projekt értékelése.....	8
6.1	A feladat értékelésének felhasználó oldali szempontjai.....	8
6.2	A feladat értékelésének technikai szempontjai.....	8
6.3	Projekt megvalósításának piaci jellegű értékelése .....	8
7	Projekt adatlap.....	9

## 2 Bevezetés

### 2.1 A feladat címe

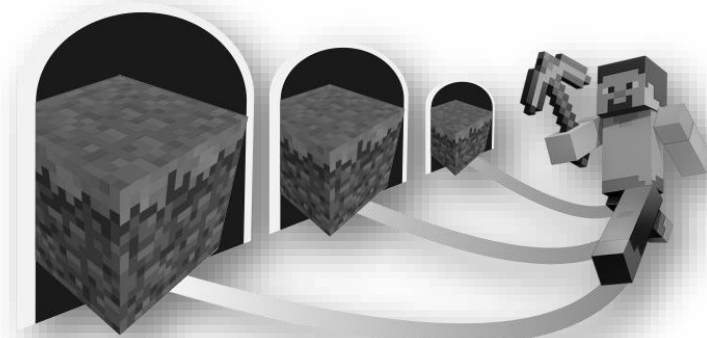
Minecraft Router

### 2.2 A feladat rövid ismertetése

Több Minecraft szerveret is üzemeltetsz otthon? Az otthoni hálózathoz több port is nyitva van az internet felé? Állj! Törekedj a biztonságra! Használj Minecraft Routert!

Otthon üzemeltetett szerverek esetén a hálózat védelme legalább annyira fontos, mint vállalati környezetben, ezért mindig a lehető legkevesebb portot szeretnénk kinyitni a szabad internet felé. Amíg webes szolgáltatások esetében egy egyszerű proxy futtatásával elegendő egyetlen port a belső rendszerek eléréséhez, addig a TCP protokollal kommunikáló szoftverek már fejfájást okozhatnak. Erre nyújt megoldást a Minecraft Router egy egyszerű trükkel.

A feladat megoldása során egy olyan alkalmazás stacket kell készíteni, amivel az internet irányából egyetlen porton érkező felhasználókat el tudjuk irányítani a megfelelő Minecraft szerver felé. A felhasználók egy weboldalon választhatják ki, hogy melyik szerverre szeretnének csatlakozni. A weboldal rendelkezik admin felülettel, ahol a felhasználók kezelése mellett, a különböző szerverekhez való hozzáférést is lehet szabályozni.



## 3 Elvárások a feladattal kapcsolatban

### 3.1 Operációs rendszer, környezet

- Windows 10+
- Linux
- Böngésző

### 3.2 Felhasználható programozási nyelv

- Nincs megkötés

### 3.3 Megoldás formátuma

- Forráskód állományok
- Teljes projekt környezet
- Forráskód dokumentáció
- Üzemeltetői dokumentáció (odt/docx és pdf formátumban)
- Felhasználó dokumentáció
- Erőforrás terv és munkaidő nyilvántartás

### 3.4 Szoftverfejlesztés

A feladat egy olyan webalkalmazás készítése, amely a felhasználóknak lehetőséget biztosít arra, hogy kiválasszák melyik szerverre szeretnének csatlakozni. A felhasználókat előzetesen egy rendszeradminisztrátor veszi fel a rendszerbe, ahol a szerverek listájából kiválasztja, hogy az adott felhasználó mely szerverek közül választhat majd. Az adminisztrátor felületet hálózati megszorításokkal kell védeni, hogy ne legyen elérhető a szabad internet irányából. Minden felhasználóhoz egy egyedi linket kell generálni, a link azonosítja a felhasználót, nincs szükség felhasználónévvvel és jelszóval történő autentikációra.

A fejlesztés közben be kell tartani a Clean Code alapszabályait.

Az elkészült megoldásnak maradéktalanul meg kell valósítania az 4-es fejezetben megfogalmazott követelményeket. Amelyik követelmény nincs pontosan definiálva, azzal kapcsolatban a megvalósítás során a fejlesztő szabad kezet kap. Fontos viszont, hogy a választott megoldás megfelelő színvonalú legyen mind felhasználói, mind fejlesztői szempontból.

## 3.5 Modulok

A projekt keretében történő megvalósítás egy lehetséges felbontási lehetősége az alábbi:

- Adatbázis tervezés, kivitelezés, DB interfész megírása
- Proxy szerver telepítés és konfiguráció
- Felhasználóbarát frontend (GUI) tervezése, kivitelezése
  - Program felület, grafikai elemek
  - Menürendszer
  - Szerver lista/adatlap
  - Felhasználólista/adatlap
- Backend osztályok megírása, amelyek a program logikáját valósítják meg
  - Adatbázis kezelése
  - Felhasználóhoz tartozó proxy konfiguráció kezelése
  - A felület kiszolgálása adatokkal
- Projektvezetéssel kapcsolatos dokumentáció, nyilvántartások vezetése, feladatok összehangolása, felhasználói dokumentáció elkészítése, tesztelés.

Lehetőség szerint a fejlesztői dokumentációkat minden esetben a ténylegesen fejlesztést végző projektagok készítsék el. A felhasználói dokumentáció külön egységet képezhet, érdemes a teszteléssel összekapcsolni a megfelelő minőség biztosítása érdekében

## 4 Specifikáció

A feladat megoldásának első lépéseként készíteni kell egy architektúrális ábrát.

### 4.1 Megjelenés

A webalkalmazás admin felületének csak asztali böngészőben kell megfelelően megjelennie, míg a felhasználóknak elérhető felületet mobil és asztali böngészőkre kell optimalizálni.

### 4.2 Funkciók

- Szerver választó (felhasználó felület):  
A felhasználó a saját linkjének megnyitásakor a hozzárendelt szervereket látja, melyek közül kattintással tud választani.  
A kattintás hatására a backend a proxy konfigurációs állományait úgy módosítja, hogy a felhasználó majd a kiválasztott szerverhez csatlakozzon.
- Adminisztrátori felület:  
A felületen két lista található, egy szerver-, és egy felhasználólista.
- Szerverlista:  
Lehessen új szervert felvenni a listára. Egy szerverhez tartozó minimális adat az IP címe és egy név.
- Felhasználólista:  
Lehessen új felhasználót felvenni a listára. A felhasználóhoz tartozhat név, amivel az adminisztrátor azonosítani tudja, illetve azon szerverek listája, amihez hozzá van rendelve.

### 4.3 Proxy

A felhasználók megfelelő szerverre való irányítására egy proxy-t kell használni. Nem szükséges saját proxy-t fejleszteni, használható bármelyik interneten fellelhető kész megoldás is, pl.: NGINX, Traefik, HAProxy, Tyk. A teljes alkalmazás stack maximum 2 porton engedélyezheti a kommunikációt a külvilág felé.

## 5 Dokumentáció

### 5.1 Erőforrásterv, munkaidő-nyilvántartás

A specifikáció birtokában a projekt résztvevői készítsenek erőforrástervet. Ez tartalmazza a feladatban részt vevő projektagokat, akik legyenek hozzárendelve a tervezés során azonosított részfeladatokhoz. Minden részfeladat mellé kerüljön egy munkaidő ráfordítási becslés munkaóraban számolva. Ezt a tervet a tényleges fejlesztés előtt le kell adni. A feladat megoldása során az elvégzett munkáról készüljön nyilvántartás részfeladatonként és személyenként a tényleges munkaórák számának megjelölésével. A projekt végén a két dokumentum összehasonlításra, az eltérések elemzésre kerülnek.

### 5.2 Üzemeltetői dokumentáció

Az üzemeltetői dokumentáció célja, hogy a rendszer üzemeltetőinek támogatást adjon a termék üzemeltetésének elsajátításához. Tartalmazza többek között a rendszer architekturális felépítését (alkalmazás stack elemei és azok közti kapcsolat leírása), az alkalmazás fordításához, fejlesztéséhez és futtatásához szükséges követelményeket, technológiákat, továbbá a konfigurációs állományok leírását (ha vannak).

A dokumentációnak a feladat bonyolultságától függő hosszúságúnak kell lennie, maximális terjedelem nincs meghatározva.

### 5.3 Forráskód dokumentáció

A fontosabb függvények és osztályok előtt szerepelnie kell megjegyzéseknek, melyeknek tartalmazniuk kell az azt követő metódus rövid szöveges – akár magyar nyelvű – leírását. A forráskód dokumentációt a munka során folyamatosan kell készíteni.

### 5.4 Felhasználói dokumentáció

Az alkalmazás használatának részletes bemutatása képernyőképekkel, funkciók pontos leírásával.

## 6 A projekt értékelése

### 6.1 A feladat értékelésének felhasználó oldali szempontjai

A működő alkalmazás tesztelése alapján az alábbiak a legfontosabb jellemzők:

- Kiírást teljes egészében lefedő funkcionalitás
- Kényelmes használat
- Igényes felhasználói felület
- Stabil működés
- Igényes felhasználói dokumentáció

### 6.2 A feladat értékelésének technikai szempontjai

Informatikai szakmai szempontból a megoldás értékelésének alapja:

- Kódkép, a kód tisztasága, kommentelés minősége
- Kódolási konvenciók betartása (Clean Code)
- Dokumentációk színvonala
- Dokumentált tesztelés
- Erőforrás felhasználásának pontos nyilvántartása

### 6.3 Projekt megvalósításának piaci jellegű értékelése

A projekt lezárultával összehasonlításra kerül a kezdeti erőforrásterv, valamint a megvalósítás során dokumentált munka. Ezen dokumentumok elemzéséből levezetésre kerülnek azok a problémák, melyek a piaci környezetben jellemzően megjelennek. Végigtekintjük ezen problémák okait, következményeit, lehetséges elkerülésüknek vagy hatásuk mérséklésének módjait. A jellemző hibák ebből a megközelítésből:

- Határidő csúszása
- Nem megfelelő minőség
- Hiányos, vagy elmaradó tesztelés
- Használhatatlan, pontatlan dokumentáció
- Pontatlan erőforrás becslés
- Aránytalanul magas önköltség
- Az elkészült termék továbbfejlesztésének, karbantartásának nehézségei

A fentiek értékelésén túl fejlesztői szemszögből elemezzük a megvalósítás tapasztalatait, a lehetséges továbbfejlesztés, átalakítás, támogatás kérdéseit és piaci lehetőségeit.



## 7 Projekt adatlap

Projekt neve: Minecraft Router

Feladat rövid ismertetése: Felhasználók megfelelő szerver felé irányítása.

Specifikációt összeállította: Selyem Ádám