Smart manager

Program Specifikáció

Készítette:

Magyar Márk, Nagy Huba



Tartalomjegyzék

[2. Bevezetés 4](#_Toc209007988)

[2.1 Rövid projekt bemutatás 4](#_Toc209007989)

[2.2 Témaválasztási magyarázat 4](#_Toc209007990)

[2.3 Mi célt szolgál 4](#_Toc209007991)

[2.4 Funkciók és extrák 4](#_Toc209007992)

[2.5 Célközönség 5](#_Toc209007993)

[3. Fejlesztőkörnyezet és hardver - Fejlesztői Dokumentáció 6](#_Toc209007994)

[3.1 A fejlesztéshez használt főbb eszközök 6](#_Toc209007995)

[3.2 Fejlesztésre használt számítógép 6](#_Toc209007996)

[3.3 UI/UX tervezés – Figma 6](#_Toc209007997)

[3.4 Technikai konzultáció 6](#_Toc209007998)

[3.5 Tárolt adatok 7](#_Toc209007999)

[4. Elvárások a feladattal kapcsolatban 8](#_Toc209008000)

[4.1 Operációs rendszer, környezet 8](#_Toc209008001)

[4.2 Felhasználandó programozási nyelvek / keretrendszerek 8](#_Toc209008002)

[4.3 Megoldás formátuma 8](#_Toc209008003)

[4.4 Szoftverfejlesztés 8](#_Toc209008004)

[4.5 Modulok 8](#_Toc209008005)

[5. Szoftver specifikáció 9](#_Toc209008006)

[5.1 Megjelenés 9](#_Toc209008007)

[6. Dokumentáció 12](#_Toc209008008)

[6.1 Erőforrás-terv, munkaidő nyilvántartás 12](#_Toc209008009)

[6.2 Technikai dokumentáció 12](#_Toc209008010)

[6.3 Forráskód dokumentáció 12](#_Toc209008011)

[7. Projekt adatlap 13](#_Toc209008012)

[7.1 Projekt neve 13](#_Toc209008013)

[7.2 Feladat rövid ismertetése 13](#_Toc209008014)

[7.3 Programozási nyelvek / Keretrendszerek 13](#_Toc209008015)

[7.4 Specifikációt összeállította 13](#_Toc209008016)

# Bevezetés

## Rövid projekt bemutatás

A Smart Manager egy projekt oktatás rendszer. A szoftver támogatja az általános felhasználást, illetve akár tanárok és diákok közötti hatékony együttműködést is elősegítheti. A rendszer célja, hogy biztosítson egy átható megoldást a projektek egy helyen történő kezelésére.

## Témaválasztási magyarázat

Azért választottuk ennek a platformnak a megvalósítását, mivel több tanár is számolt be nehézségekről. Többek között a diákok nagy száma, valamint a közös munkát támogató platformok sokfélesége is kihívást jelentett. Hiszen vegyük alapul például van 4 darab 2-2 fős csapat osztályonként, az egyik csapat discordot és githubot használ, ahol követhető ki mit csinált és mikor, egy másik csapat pedig használt messengert, és drive-ot, ahol nem lehet olyan egyszerűen vissza követni, hogy ki, mikor, és mit változtatott.

## Mi célt szolgál

A Smart Manager célja egy olyan projektmenedzsment rendszer létrehozása, amely egyszerre segíti a felhasználók együttműködését és az adminisztrációs munkálatokat, átláthatóvá teszi a feladatokat és a statisztikákat. Remek eszköz bármilyen célú felhasználásra, mivel például ha egy cég minket választ, tudják kezelni dokumentumaikat, illetve egyéb munkafolyamatokat és az alkalmazottakat követni, hogy ki mikor mit végzett egy adott projekt során.

## Funkciók és extrák

Három fő funkciót egyesít: verziókezelés, feladatmenedzsment és valós idejű kommunikáció. A verziókezelő rész lehetővé teszi a projekthez tartozó fájlok rendszerezett tárolását és verziókövetését. A feladatmenedzsment egy Kanban tábla segítségével támogatja a csapatok munkaszervezését és nyomon követését. A beépített chat modul biztosítja a gyors és egyszerű kommunikációt a csapattagok között, így minden fontos eszköz egyetlen felületen érhető el.

## Célközönség

A platform elsődleges célközönsége a diákok és tanárok, akik egyszerű, átlátható eszközt keresnek közös projektek, beadandók és csapatfeladatok megszervezésére. Segíti a feladatok kiosztását, a közös dokumentum- és kódrendszerezést, valamint a gyors kommunikációt. Emellett a rendszer hasznos lehet kisebb fejlesztői csapatoknak, hobbi projektekhez vagy nonprofit szervezeteknek is, ahol fontos a könnyen kezelhető, integrált munkakörnyezet.

# Fejlesztőkörnyezet és hardver - Fejlesztői Dokumentáció

## A fejlesztéshez használt főbb eszközök

* IDE: Visual Studio 2022
* Célplatform: Windows, Android, IOS
* Verziókezelés: Git (GitHub)

## Fejlesztésre használt számítógép

|  |  |  |
| --- | --- | --- |
| Specifikáció | Minimum | Maximum |
| RAM: | 16GB DDR4 | 16GB DDR4 |
| OS: | Windows 10 64bit | Windows 11 64bit |
| GPU: | AMD Radeon RX 570 8gb | Geforce rtx 2070 super 6gb |
| CPU: | AMD Ryzen 5 2600 | Intel Core i7-10970H |

## UI/UX tervezés – Figma

A program felhasználói felületének és képernyőelrendezésének megtervezésére a Figma nevű online UI/UX eszközt használtuk. Ez lehetővé tette, hogy már a fejlesztés megkezdése előtt vizuálisan átgondolt terveket készítsünk a menürendszerről, szövegkörnyezetről, valamint az interaktív elemek elhelyezkedéséről. A Figma használatának köszönhetően el tudtunk kerülni több hiba lehetőséget is. Mivel meg volt tervezve egy alap koncepció így tudtuk csökkenteni a felesleges újra tervezések számát és hatékonyabban dolgoztunk.

## Technikai konzultáció

A közös munka során megbeszélésekre és egyeztetésekre a Discordot használtuk hiszen rendkívül hasznos és széleskörben elterjedt alkalmazás. Ez biztosította, hogy meg tudjuk egymással osztani hívás közben a képernyőnket és élőben tudjuk követni min dolgozik a csapattársunk. Ezen felül személyes találkozások alkalmával is tudtunk együtt dolgozni, amely segítette előrehaladásunkat.

## Tárolt adatok

A programhoz kapcsolódó összes változó, adat, ide tartoznak többek között:

* Felhasználói adatok (felhasználónév, jelszó, email cím)
* Projektmunka adatai (név, tagok)
* A feltöltött munka adatai (fileok - Pl: css, js, html, cs, xml, txt…stb.)
* Valós idejű kommunikáció (üzenetek)
* Statisztika (adott projektben munkaóra, stb.)
* Aktivitás (ki mikor volt utoljára elérhető)
* To-Do táblázat adatai (Határidő, feladat státusz: Befejezett, folyamatban, Késés, ha későn adja le a User akkor számolja mennyi idővel később adta le)
* Jogosultságok és szerepkörök (pl: Admin, tanár, diák, projektvezető)
* Jogosultság szint: be lehet állítani, hogy ki mit láthat és mit módosíthat létrejött csapat(ok) adatai.

# Elvárások a feladattal kapcsolatban

## Operációs rendszer, környezet

* A rendszer többplatformos működésre lett tervezve.
* Reszponzív web alkalmazás, amely modern böngészőkben (Chrome, Edge, Safari) és mobil eszközökön is működik, webböngészőn keresztül, egyaránt androidon és iphone-on (Safari, Google Chrome).
* Asztali kliens: Windows környezetre készített WinUI3 alkalmazás.

## Felhasználandó programozási nyelvek / keretrendszerek

* Backend: ASP.NET Core
* Frontend: Vue.js
* Asztali kliens: C# (WinUI3 / Avalonia)
* Adatbázis: SQL (PostgreSQL)
* Valós idejű kommunikáció: Websocket (Socket.IO)

## Megoldás formátuma

A végső megoldás kettő komponensből áll:

1.Webes frontend és Asztali kliens –felhasználó számára projektek kezelése / adminisztrátor számára statisztikák és tagok kezelése.

2.Backend REST API – a központi logika és adattárolás.

A teljes forráskód, dokumentációval együtt GitHub repository-ban kerül tárolásra, verzió követéssel.

## Szoftverfejlesztés

A fejlesztés során Visual Studio-t, Postman-t és Figma-t használtunk, a kommunikáció Discordon zajlott. A tiszta kód elveit követtük, és folyamatosan teszteltük a rendszer stabilitását.

## Modulok

* Felhasználókezelés (bejelentkezés, regisztráció, jogosultságok).
* Projektfeladat-kezelés (Kanban board, státuszok).
* Statisztikák és riportok (munkaóra, feltöltések száma, stb.).
* Adminisztrációs felület (tanári asztali kliens).

# Szoftver specifikáció

## Megjelenés

Felhasználóknak/diákoknak amit tartalmazni fog:

1. Belépés

* bejelentkezés / regisztráció -> e-mail + jelszóval.

2. Főoldal

* Közelgő határidők (beadandók).
* Saját státusz (hány feladat folyamatban, hány kész).
* Legutóbbi közlemények.

3. Feladatkezelés

* Kapott feladatok listája.
* Feladat beadása (fájl feltöltés → pl. .zip, .docx, .cs).
* Saját beadások állapotának követése (pl. „Folyamatban”, „Elküldve”, „Értékelve”, „Elkésve”).
* Határidő figyelmeztetés (pl. piros kiemelés, ha lejárt és időzítő indul, hogy mióta nincs leadva).

4. Csapatmunka

* Kanban tábla: „Teendő – Folyamatban – Hiányzik – Kész”.
* Feladatkártyák: név, felelős, határidő.
* Közös fájlok feltöltése.
* Változásnapló: ki mikor adott hozzá/módosított valamit.

5. Chat modul

* Egyéni chat felhasználó ↔ felhasználó.
* Csapatchat (projekt szinten).
* Adminisztrátor-felhasználó chat (pl. kérdés beadandóval kapcsolatban).

6. Statisztika / Visszajelzések

* Beadott feladatok száma és státusza.
* Értékelések megtekintése.
* Grafikon: teljesítési arány, késésekkel együtt.

7. Beállítások

* Dark/Light mód.
* Profil Adatok (e-mail, jelszó).

Adminisztrátor/Tanároknak, amit tartalmazni fog:

1. Belépés

* Tanár/Admin bejelentkezés (magasabb jogosultságokkal).

2. Főoldal

* Legutóbbi beadott feladatok.
* Figyelmeztetések (pl. „5 fhsz. nem adta le a feladatot időre”).
* Statisztikai összegzés (átlag pontszám, teljesítési arány).

4. Feladatkiosztás

* Új feladat létrehozása (név, leírás, határidő, csatolmány).
* Feladat kiadása.
* Határidők követése → listázás, ki adta le időben.

5. Értékelés

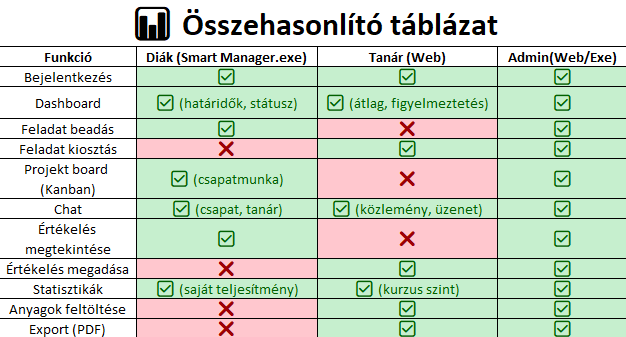
* Beadott feladatok megtekintése (ki mikor adta le, milyen fájlt adott le).
* Pontszám / jegy megadása.
* Szöveges visszajelzés írása.
* Automatikus státusz frissítés (pl. „Értékelve”).

6. Statisztikák és riport, osztályátlag megjelenítése.

* Legjobb és legrosszabb teljesítők listája.
* Hiányzó beadások, késések kimutatása.
* Exportálás PDF-be.

7. Kommunikáció

* Üzenet küldés.
* Egyéni üzenet felhasználóknak/diákoknak.
* Visszajelzés beadott munkákhoz közvetlenül.

Összefoglalva Excel táblázatban:

# Dokumentáció

## Erőforrás-terv, munkaidő nyilvántartás

* Csapatlétszám: 2 fő fejlesztő.
* Munkamegosztás: egyik fő a backend + desktop fejlesztésért felel, a másik fő a frontend + UI megvalósításáért.
* Munkaidő: heti 8–10 óra fejenként, összesen kb. 150–160 munkaóra.
* Ütemezés: 8 hét, heti mérföldkövekkel (lásd Gantt-diagram).

## Technikai dokumentáció

* Adatbázismodell: ERD diagram, SQL dump.
* Rendszerarchitektúra: backend, frontend és desktop komponensek kapcsolata.
* API dokumentáció: Swagger/OpenAPI specifikáció a végpontokról.

## Forráskód dokumentáció

* Kommentek: tiszta kód elvek, metódusonkénti rövid leírás.
* README fájl: telepítési és futtatási útmutató.
* Kódfelépítés: modulok és mappastruktúra bemutatása.
* Tesztdokumentáció: futtatott tesztek és eredmények.

# Projekt adatlap

## Projekt neve

Smart Manager

## Feladat rövid ismertetése

Projekt menedzser, amely kiszolgál egy teljes projektfolyamat lebonyolításához szükséges minden igényt.

## Programozási nyelvek / Keretrendszerek

C#, HTML, CSS, Javascript, SQL, Vue.js, ASP.NET Core

## Specifikációt összeállította

Magyar Márk József, Nagy Huba Kende