AFP Minta

A tárgy célja: Szemléltetni a diákokkal a manapság használt rendszereket az alkalmazásfejlesztés szempontjából, illetve megmutatni, hogy hogyan működik egy “nagyobb” projekt munkavégzési folyamata. Ezen kívül a csapatban való munkaképesség fejlesztése sem másodlagos szempont.

Verziókövetés/GIT: Megismertetni a diákokat a Git verziókövető rendszerrel. Több haszna is van, például hardveres probléma esetén nem bukjuk a munkánkat, bárhol elérhetjük őket és ezeket commitok alapján verziókként nyomon is tudjuk követni, ha egy újabb verzióban valami esetleg nem működik.

Feladatkövetés/Trello: Megismertetni a diákokat a Trello feladatkövető rendszerrel. Ennek is szintén több haszna van, mivel így folyamatosan látjuk, hogy milyen feladatok vannak, amiket el kellene vállalni valakinek, illetve nyomon tudjuk követni, hogy ki min dolgozik, milyen feladatok vannak folyamatban, milyen bugok vannak a rendszerben, mit tesztelnek éppen és ténylegesen mely részek vannak készen. Ez természetesen függ, hogy a trellonak milyen oszlopokkal látjuk el és hogy az adott projektcsoport hogyan kezeli azt.

Dokumentumok

Követelmény Specifikáció: A követelmény specifikáció lényegében egy olyan dokumentum, amiben a megrendelő által elkészítésre váró szoftver/keretrendszer leírása található.

Funkcionális Specifikáció: A követelmény specifikáció kidolgozottabb formája. Ebben a dokumentumban sokkal részletesebben le van írva a kívánatos szoftver/rendszer, hogy a fejlesztők ténylegesen átlássák és megértsék, hogy valójában mit is kell csinálniuk.

Rendszerterv: A rendszerterv egy írásban rögzített specifikáció, amely leírja, hogy mit (rendszer), miért (rendszer célja), hogyan (terv), mikor (időpont) és miből (erőforrások) akarunk a jövőben létrehozni. Fontos, hogy reális legyen, azaz megvalósítható lépéseket írjon elő. A rendszerterv hasonló szerepet játszik a szoftverfejlesztésben, mint a tervrajz az építkezéseken, tehát elég részletesnek kell lennie, hogy ebből a programozók képesek legyenek megvalósítani a szoftvert.

GIT

Mi a GIT: A git egy verziókövetést lehetővé tevő szolgáltatás. A leghíresebb git alapú portál a GitHub, de léteznek alternatívák, illetve akár saját magunknak is létrehozhatunk egy saját git szervert. Léteznek alternatívák is a gitre: Subversion, CVS vagy Mercurial.

Miért jó a verziókövetés: Röviden azért, mert ha egy komolyabb projektben dolgozunk és valami elromlik, legyen hardveres vagy szoftveres vagy éppen a fejlesztett program egy új verziója nem úgy viselkedik ahogy kellene akkor egyszerűen pár kattintással tudunk rollbackelni és nem kell fogni a fejünket, hogy mit kell visszaállítani ahhoz hogy jól menjen úgy ahogy eddig. Illetve, ha jól működik akkor szimplán fel tudjuk erre a szerverre tölteni a fájljainkat és bárki, akinek hozzáférése van a repositoryhoz onnantól kezdve eléri.

Mi az a repository: A repositoryt(röviden "repó") úgy kell elképzelnünk, mint egy dosszié egy nagy irattárolóban, vagy egy szoba egy motelben. A szobához mindenki hozzáfér, akinek kulcsa van hozzá, bármit be tud vinni illetve ki tud vinni, de bármit is végez ott azt fel kell jegyeznie. Ha valaki utána bemegy úgy találja, ahogy otthagyta az előtte lévő.

Érdekes: Az első komolyabb verziókövető szolgáltatás 2005 körül jelent meg, viszont ez előtt is használtak verziókövetést, floppyn és egyéb adattárolókon egy biztonsági szobában. Kicsit kényelmesebb ehhez képest a gitelés.

Hogyan kell elképzelni a működését: Valaki létrehoz egy repót amibe meg tud hívni más embereket a hozzáférésért. Ha valaki feltölt egy kliensen keresztül egy fájlt (commit-ol majd push-ol) ez a fájl meg fog jelenni a repón belül mindenki számára, akinek hozzáférése van. Ha valaki ezt a fájlt felül akarja írni akkor ugyan úgy csak fel kell töltenie azt a fájlt, viszont a Git eltárolja az előző verziót is, így elérhetővé válik az összes eddig commitolt változat, de első látszatra mindig csak a legaktuálisabb verzió lesz jelen(Ha valaki felülírja a fájlt és te megpróbálod leklónozni a repót az első alkalommal, vagy simán letölteni a változásokat(pull) akkor az aktuális verzió lesz csak jelen, az előző verzió kódja egy korábbi commitként fog csak szerepelni a logban). Ami gyakran nagy előnye a gitelésnek, hogy ha például felülírsz egy meglévő fájlt, akkor logban minden változást látsz kigyűjtve (Ezek a sorok törlődtek az előző verzióból, viszont ezek újak. Ezt a - és + jelekkel szokta jelezni). Ami nagyon fontos, hogy minden commitnál egy lényegretörő címmel és leírással lásd el a verziód, mert a későbbiekben sokat segíthet, ha leírod mi módosult és nem csak egy "updated xy.cs on 2019. 09. 08"-at fogsz látni 0 leírással.

Milyen Git kliensek vannak: Számos git alapú kliens van, amik általában működésüket tekintve megegyeznek, de mégis eltérnek kissebb nagyobb mértékben egymástól. Ilyen kliens például a GitHub desktop, a GitKraken, SourceTree vagy a TortoiseGit.

Scrum
Gyorstalpaló röviden

Célja: Egy adott projekt lebontása kisebb, értelmezhető részekre, úgy, hogy lehetőleg minden egyes apróbb szegmens működő részt adjon hozzá a projekthez.

Szerepek:
- **Dev Csapat:** A tényleges fejlesztői csapat, általában 4-7 emberből áll.
- **Scrum Master:** A csapat részét alkotja. Feladata, hogy a problémákra megoldást találjon, a csapatot hatékonyabbá és produktívabbá tegye, illetve védi a csapatot a külső nyomástól. Emellet Retrospektíveket csinál.
- **Product Owner:** A feladatokat definiálja, az elkészült munkákat ellenőrzi és elfogadja. Kapcsolatot tart a külső ügyfelekkel, Ő az egyedüli felelős a termékért és a hétköznapi nyelvű felhasználói igényekből ő készít user storyt. Például: “Ha megnyomom a gázpedált, akkor menjen az autó.”

Sprint Rendszer:
- Általában 1-2 hét, amikor a csapat gyorsított módban dolgozik. Mindig egy meetinggel indul, amin mindenki, aki a fejlesztéssel foglalkozik, az részt vehet. Közösen meghatározzák a maximum kapacitási számot, ami azt jelenti, hogy egy pontrendszerben mindenkinek van egy bizonyos teljesítmény indexe. Ezeket a pontokat később felhasználják a feladatok nehézségének meghatározására úgy, hogy az összkapacitásból maradjon valamennyi esetleges hibák, csúszások érdekében. Mindig a dev csapat választja ki és mondja meg, hogy mit vállal be, a product owner pedig próbál minél több feladatot adni nekik. Egy meeting maximum 4 óráig tart, ezután kezdetét veszi a munka.
- Minden nap van egy stand up meeting, ahol a csapattagok elmondják, hogy mivel, hogyan haladtak, mi volt a pozitívum, negatívum és hogy milyen akadályokat lát. Cél, hogy az összes probléma azonnal kiderüljön, megmaradjon a motiváció és a scrum master tisztában legyen azzal, hogy éppen hogyan állnak a csapattagok, a product owner pedig látja a projekt előrehaladását, továbbá a csapattagok friss szemmel dolgozhassanak aznap.
- A feladatokat egy feladatkövető rendszeren rögzítjük azért, hogy bárki láthassa, akinek érdekében áll, hogy nyomon kövesse a projekt előrehaladását. A rendszerben láthatjuk, hogy mely feladatok vannak folyamatban, illetve melyek azok, amikhez még nem kezdtek hozzá vagy már elkészültek. A folyamatban lévő feladatok limitálva vannak, maximum 4-5 lehet csapatszinten. Ez azért van, hogy véletlenül se fordulhasson elő az, hogy több feladatba is belekezd a csapat, de egy sem készül el.
- A sprint végén van egy kötelező review, ahol ránéz egy a rangsorban magasabban vagy azonos szinten álló ember az elkészített munkára és azt értékeli, elmondja, hogy milyen hibákat lát benne, ezzel is biztosítva azt, hogy mindenképpen jó, hibamentes és olvasható munka kerüljön ki a fejlesztők kezéből.
- A legeslegvégén van egy retrospektív, azaz meeting, ahol átbeszéljük, hogy mik a tapasztalatok a még hatékonyabbá válás céljából. Pozitívum,

negatívum, minden számít és nagyon fontos! Itt derül ki, hogyan változik a csapattagok kapacitása és ezzel együtt az összkapacitás.

- A sprint ciklikusan halad, azért, hogy 2 hetente valamit ki tud “adni” fejlesztés közben.

Követelmény Specifikáció

1. Áttekintés

- Ebben a fejezetben röviden ismertetni kell a projektünk egészét. Milyen technológiákat szeretnénk alkalmazni, hogyan fog működni az alkalmazásunk stb.

- **Példa:** *"Az alkalmazás célja játékos módú iskolai tanulmányokat kiegészítő tanulási lehetőség biztosítása általános iskolásoknak. Az alkalmazás rendelkezik Web és Android felülettel, mindkettő módon elérhető az összes felhasználói funkció. A tananyagok tantárgyak, azon belül pedig témakörök szerint vannak osztályozva. A témakörökben a diákok tananyagot és ahhoz tartozó teszteket (kiegészítős, feleletválasztós, összekötős, közelítős) találnak. Ezek megoldására idő és pontosság alapján pontszámot kapnak a tanulók. Az eredmények megjelennek az adott feladat mellett, illetve az összesített pontszám megjelenik egy külön menüpontban. (összesített pontszámok)."*

2. Jelenlegi helyzet

- Ennek a fejezetnek a feladata kifejteni, hogy miért van szükség az alkalmazásunkra. Érdemes minél lényegretörőbbnek lenni benne, minél több pontban ecsetelve a szükséget.

- **Példa:** *"Az oktatásért felelős szervek hierarchikus rendszerében legalul foglalnak helyet az iskolák. Ebből adódóan nem azok alakítják ki a tanrendet akik a tanulók érdekeit tartják szem előtt. Jelenleg az oktatási rendszer: Szigorú a diákokhoz; A tanárok nem tudnak párhuzamosan foglalkozni a diákokkal; Nincs lehetőség a diákok egyéni igényeihez igazodni; lexikális tudásközpontú, stresszes, nem motiváló, nem játékos; elméleti alapú, nagyon kevés gyakorlattal; XX. századi, elavult módszereket használó. Ezek miatt a mai oktatás alapvetően nyomasztó és demotiváló hatással bír."*

3. Vágyálom rendszer

- A vágyálom rendszer azért felelős, hogy kifejtsük benne mit szeretnénk célul a programunkban a 100%-ban ideális esetben. Ilyen-olyan feature-ök jelenléte, és ideális állapotuk stb.
- **Példa:** *"A projekt célja egy olyan rendszer, ami játékos tanulás, gyakorlás lehetőségét biztosítja. A rendszer elérhető több platformon, weben és androidon is. Regisztrációt követően több típusú feladat közül választhat a felhasználó. Látványos, színes felülettel rendelkezik a program, hogy felkeltse a felhasználók figyelmét. A szórakoztatóbb tanulás érdekében játékos elemeket tartalmaznak az egyes feladatok. A rendszer lehetőséget nyújt a felhasználók teljesítményének tárolására (toplista), ennek segítségével másokkal is összemérhetik a tudásukat. A rendszerben szükséges egy pontozási rendszer, amely a helyesen megválaszolt feladványok után adja a felhasználónak a pontokat. Ez a pontozási rendszer optimális esetben függ a feladatokhoz meghatározott időtől is.A rendszernek van egy admin felülete is, ahol az admin fiókkal bejelentkezett felhasználó fel tudja tölteni a feladványokat. "*

4. Funkcionális követelmények

- A funkcionális követelményeknél azt kell leírnunk, hogy a program bizonyos részegységeihez milyen funkciók tartoznak. Milyen felhasználói, adminisztrációs, egyéb funkciók vannak stb.

5. Rendszerre vonatkozó törvények, szabványok, ajánlások

- Talán az egyik legkényesebb része a dokumentumnak. A cím magáért beszél, le kell írnunk a programunkra vonatkozó jogszabályokat, mely szabványok szerint szeretnénk elkészíteni.

6. Jelenlegi üzleti folyamatok modellje

- Bővebben kifejteni azt a folyamatot, amit a programunk szeretne leváltani vagy kibővíteni. A példa alapján ez az iskolai tanulás folyamata.
- Példa: "A mai világban az oktatás nem használja ki a már meglévő *technológiákat arra, hogy a tanulást sokkal szórakoztatóbbá és interaktívabbá tegye. A jelenlegi világban a fiatalok egyre kevésbé hajlandóak a "klasszikus" módon tanulni, ezért a különböző oktatási intézmények alternatív módszereket keresnek. Jelenleg a diákok tankönyvekből tanulnak és papír alapon adnak számot tudásukról, amely a XXI. században elavultnak számít. Ez rengeteg nyomdai és nyomtatási költséget jelent. Az oktatóknak rengeteg időt elvesz az idejéből a dolgozatok egyesével való kijavítása. Illetve a dolgozatok megírása papíron is sokkal időigényesebb, mintha különböző alkalmazásokat használnánk a diákok számonkérésére. "*

7. Igényelt üzleti folyamatok

- A 6. pontban leírt folyamatot hogyan szeretnénk leváltani vagy kibővíteni.
- **Példa:** *"A megrendelő a fő oldalon akar bejelentkezni (felhasználónév, jelszó), valamint a regisztrációt megkezdeni, mely egy új oldalon folytatódna (felh., jelszó, e-mail). Bejelentkezést követően lehetőséget, hozzáférést kell adni szerepkörtől függően az alkalmazás funkcióihoz. A feladatokat a tanár állíthatja össze, ezért ezeket részekre kell szedni. Kvíz mely konkrét választ vár. Kvíz, mely több válasz közüli lehetőséget kínál, ezen belül 1 vagy több helyes válasszal. Rajz feladat, és ezek tetszőleges kombinációja lenne egy feladatsor."*

8. Követelménylista (KÖTELEZŐ!)

- A programozás szempontjából talán a legfontosabb része a dokumentumnak. Itt kell leírni azt, hogy milyen funkciókkal kell rendelkeznie a programunknak, ezeknek milyen al-funkciói vannak.
- **Példa:**

Modul	ID	Név	v.	Kifejtés
Jogosultság	K1	Bejelentkezési felület	1.0	A felhasználó az email címe és a jelszava segítségével bejelentkezhet. Ha a megadott email cím vagy jelszó nem megfelelő, akkor a felhasználó hibaüzenetet kap.
Jogosultság	K2	Regisztráció	1.0	A felhasználó a felhasználói nevének, email címének és jelszavának megadásával regisztrálja magát. A jelszó tárolása kódolva történik az adatbázisban. Ha valamelyik adat ezek közül hiányzik vagy nem felel meg a követelményeknek, akkor a rendszer értesíti erről a felhasználót.
Jogosultság	K3	Jogosultsági szintek	1.0	- Admin : Rendszerhozzáférés, feladatok feltöltése, felhasználók / szerepkörök módosítása. - Tanár : Feladatok feltöltése / létrehozása, elektronikus napló hozzáférés, órarend, üzenetek. - Diák : Feladatok kitöltése / elvégzése, elektronikus napló látható, órarend, üzenetek. - Szülő: Diák órarendje, üzenetek, Diák elektronikus naplója látható.
Modifikáció	K4	Felhasználó módosítása	1.0	A felhasználó módosítani tudja saját Felhasználónevét. Ehhez szükséges a régi és az új felhasználók megadása, az új megerősítése, valamint a felhasználó jelszavának megadása.
Modifikáció	K5	Jelszó módosítása	1.0	A felhasználó módosítani tudja saját jelszavát. Ehhez szükséges a régi és az új jelszavának megadása, valamint az új megerősítése.
Modifikáció	K6	Elfelejtett felhasználónév / jelszó	1.0	Ha a felhasználó elfelejtette a felhasználónevét, vagy jelszavát akkor ezzel az opcióval egy Adminhoz tud fordulni.
Feladattípus	K7	Kvíz	1.0	Több kérdésből áll, a feladat a helyes válasz kiválasztása több lehetőség közül. A felhasználó az eltelt idő függvényében pontot kap.
Feladattípus	K8	Teszt	1.0	A teszthez hasonló, ugyanolyan több kérdésből álló feladatsor, ahol az idő számít. A kapott pontszám alapján érdemjegyet kap a felhasználó.
Statisztika	K9	Toplista	1.0	Egy lista a játékosok pontszámairól, a lista elején a legtöbb pontot elért felhasználó található.
Felület	K10	Üzenetek	1.0	A felhasználók egymást között tudnak küldeni üzeneteket, jogosultságuktól függően.

Felület	K11	Órarend	1.0	A felhasználóknak, szerepkörtől függően, van egy órarendje, ahol láthatják, melyik órák, mikor hol lesznek.
Felület	K12	Elektronikus Napló	1.0	A felhasználók itt láthatják megszerzett érdemjegyeiket, dicséreteiket, rovásaikat. Szerepkörtől függően mást látnak.
Felület	K13	Bejelentkezés	1.0	A felhasználók itt tudnak bejelentkezni a rendszerbe, probléma esetén jelszót, emailt változtatni.
Felület	K14	Teszt létrehozás	1.0	A tanári jogosultságú felhasználóknak elérhető a teszt létrehozás, amit később kiadhat diákok számára.
Jogosultság	K15	Admin felület	1.0	Felület az admin fiókkal rendelkező felhasználó számára. Tartalmaz egy felületet az új feladatok feltöltéséhez.

”

9. Riportok

- Létezik az úgynevezett szabad riport amelyben igazából csak azt a kérdést tesszük fel, hogy hogyan kellene működnie az új rendszernek. A másik típus az irányított amely inkább egy rövid válaszokra épülő kérdőívnek tűnhet.

10. Fogalomtár

- A cím magáért beszél. Itt kell leírnunk azokat a kifejezéseket, amelyek a programunkban fognak illetve ebben a dokumentumban már szerepelnek, és nem biztos, hogy érthető egy külsős embernek.
- **Példa:** *“Reszponzív felület - Mobilon, Tableten, PC-n igazodik a képernyőhöz a felület mérete, azaz több eszközön is probléma nélkül üzemelhet.”*

Funkcionális Specifikáció

1. Áttekintés

- Ebben a fejezetben röviden ismertetni kell a projektünk egészét.
- **Példa:** *”Egy olyan rendszert fejlesztünk, ami segíti a fiatalok tanulását, és a célunk, hogy a felhasználó a lehető legfrissebb tudáshoz jusson. Lehetőségük lesz a felhasználóknak játékos formában tanulni és ezáltal az oktató színesebbé, és játékosabbá teheti az óráját. Rengeteg téma közül lehet majd választani, épp ezért szinte bárki számára hasznos lesz ez az online felület. Természetesen nem csak számítógépen lesz elérhető az alkalmazás, hanem célunk hogy minél több platformon hozzá lehessen férni, legyen az tablet vagy telefon. Ez a rendszer ingyenes lesz, ezért bárki bárhonnan le tudja majd tölteni a telefonjára vagy esetleg interneten keresztül beregisztrál és máris hozzájut a legfrissebb tudáshoz. Különböző feladatok lesznek az egyes témák végén amivel a felhasználó próbára teheti a tudását. Minden ilyen feladat megoldása után az adott személy láthatja, hogy mennyi pontot szerzett, és mint ez egy vissza igazolást ad a számára, hogy mennyire sikerült elsajátítania az adott témakört.”*

2. Jelenlegi helyzet

- Kifejti, hogy miért van szükség az adott alkalmazásra, de itt már nem a kifejtés mértéke és részletessége a fontos, hanem a lényegretörő, vázlatos, egyértelmű leírása.
- **Példa:** *”A megrendelő szeretné kibővíteni az általa nyújtott oktatási szolgáltatások elérhetőségét, hogy versenytársaihoz képest így jusson piaci előnyökhöz. Egy új rendszer előállítását rendelte meg, amely interneten keresztül modern megoldásokat használva működik. A rendszer segítséget nyújt azok számára, akik valamilyen módon szeretnék tudásukat elmélyíteni egy adott területen, vagy új ismereteket szeretnének szerezni. A XXI. század megköveteli, hogy mindez hálózaton is elérhető legyen, ennek megfelelően Android alkalmazást és weboldalt is a megrendelő rendelkezésére kell bocsátani. Eddig a megrendelő csak számítógépen való gyors írás elsajátításához használt egy Stamina nevű alkalmazást. Ez a fajta tanulás megtetszett a diákoknak, és a tanárok is felismerték, hogy egy játékos oktatóprogrammal eredményesebbé, szórakoztatóbbá tehetik az oktatást, mint a hagyományos, táblára írásos módszerrel. Ezért elkezdtek keresni számukra megfelelő, létező programokat.*

Találtak többet is (pl.:Kahoot), viszont ezek nem tetszettek nekik. Egy olyan programra lenne szükségük, amellyel a diákok önállóan tudnak feladatot megoldani, és a többi játékos pontszámához képest tudják viszonyítani magukat. Mindezek mellet a megrendelőnek szüksége van a saját logojuk feltüntetésére is. Ezen okokból kifolyólag megkértek minket, hogy csináljuk meg nekik a vágyott alkalmazást, ami sokkal könnyebbé teheti számukra az oktatást."

3. Követelménylista

- Itt kell leírni azt, hogy milyen funkciókkal kell rendelkeznie a programunknak, ezeknek milyen al-funkciói vannak. A követelmény specifikációhoz képest itt sokkal részletesebben és pontosabban le vannak írva a követelmények.
- **Példa:**

Modul	ID	Név	v.	Kifejtés
Jogosultság	K1	Bejelentkezési felület	1.0	A felhasználó az email címe és a jelszava segítségével bejelentkezhet. Ha a megadott email cím vagy jelszó nem megfelelő, akkor a felhasználó hibaüzenetet kap.
Jogosultság	K2	Regisztráció	1.0	A felhasználó a felhasználói nevének, email címének és jelszavának megadásával regisztrálja magát. A jelszó tárolása kódolva történik az adatbázisban. Ha valamelyik adat ezek közül hiányzik vagy nem felel meg a követelményeknek, akkor a rendszer értesíti erről a felhasználót.
Jogosultság	K3	Jogosultsági szintek	1.0	- Admin : új feladat feltöltése - Felhasználó : feladatok kitöltése, toplista megtekintése, jelszó módosítása - Vendég : regisztráció, belépés
Modifikáció	K4	Felhasználó módosítása	1.0	A felhasználó módosítani tudja saját Felhasználónevét. Ehhez szükséges a régi és az új felhasználók megadása, az új megerősítése, valamint a felhasználó jelszavának megadása.
Modifikáció	K5	Jelszó módosítása	1.0	A felhasználó módosítani tudja saját jelszavát. Ehhez szükséges a régi és az új jelszavának megadása, valamint az új megerősítése.
Modifikáció	K6	Elfelejtett felhasználónév / jelszó	1.0	Ha a felhasználó elfelejtette a felhasználónevét, vagy jelszavát akkor ezzel az opcióval egy Adminhoz tud fordulni.
Feladattípus	K7	Kvíz	1.0	Több kérdésből áll, a feladat a helyes válasz kiválasztása több lehetőség közül. A felhasználó az eltelt idő függvényében pontot kap.

Feladattípus	K8	Teszt	1.0	A teszthez hasonló, ugyanolyan több kérdésből álló feladatsor, ahol az idő számít. A kapott pontszám alapján érdemjegyet kap a felhasználó.
Statisztika	K9	Toplista	1.0	Egy lista a játékosok pontszámairól, a lista elején a legtöbb pontot elért felhasználó található.
Felület	K10	Üzenetek	1.0	A felhasználók egymást között tudnak küldeni üzeneteket, jogosultságuktól függően.
Felület	K11	Órarend	1.0	A felhasználóknak, szerepkörtől függően, van egy órarendje, ahol láthatják, melyik órák, mikor hol lesznek.
Felület	K12	Elektronikus Napló	1.0	A felhasználók itt láthatják megszerzett érdemjegyeiket, dicséreteiket, rovásaikat. Szerepkörtől függően mást látnak.
Felület	K13	Fórum	1.0	Egy felhasználói fórum, ahol a felhasználók tudnak érdekességekről beszélni, vitatkozni.
Jogosultság	K14	Admin felület	1.0	Felület az admin fiókkal rendelkező felhasználó számára. Tartalmaz egy felületet az új feladatok feltöltéséhez.

4. Jelenlegi üzleti folyamatok modellje

Példa: *"A mai világban az oktatás nem használja ki a már meglévő technológiákat arra, hogy a tanulást sokkal szórakoztatóbbá és interaktívabbá tegye. A jelenlegi világban a fiatalok egyre kevésbé hajlandóak a "klasszikus" módon tanulni, ezért a különböző oktatási intézmények alternatív módszereket keresnek. Jelenleg a diákok tankönyvekből tanulnak és papír alapon adnak számot tudásukról, amely a XXI. században elavultnak számít. Ez rengeteg nyomdai és nyomtatási költséget jelent. Az oktatóknak rengeteg időt elvesz az idejéből a dolgozatok egyesével való kijavítása. Illetve a dolgozatok megírása papíron is sokkal időigényesebb, mintha különböző alkalmazásokat használnánk a diákok számonkérésére."*

5. Igényelt üzleti folyamatok modellje

- **Példa:** *"Azért hogy egyszerűbbé tegyük a diákok és a tanárok feladatát, létrehozunk egy programot ami a mai kornak megfelelően helyt tud állni az elektronikai világban. A tanároknak egyszerűbb lesz mert csak egyszer kell felvinniük a rendszerbe a feladatsort és a helyes válaszokat is csak egyszer kell kiválasztania. Ezáltal nem kell minden dolgozatot egyesével átvizsgálni lepontoznia és érdemjegyet adni rá, így sokkal több időt megtakaríthat. A diákoknak is sokkal jobb mert nem kell azon görcsölniük hogy milyen lett az eredmény mert a teszt kitöltése után egyből megtudják az eredményt és a hibás válaszokra is a helyes választ. Illetve tanulni is sokkal könnyebb*

nekik mert csak előkeresik a az éppen feladott leckét és már tanulhatják is és egyből ellenőrizhetik magukat. Nem kell minden egyes könyvet külön előkeresni megkeresni a fejezetet végéig lapozni. A szülők egyből értesülnek a dolgozatok eredményeiről."

6. Használati esetek

- Ez a fejezet leírja pl. melyik felhasználó milyen funkciókat tud használni.
- **Példa:** *"ADMIN: Az ADMIN beléphet mindegyik más szerepkörbe, hogy az hibamentes működését ellenőrizhesse. Az Admin(ok) feladata a rendszer problémamentes működése. Ez egyben jár azzal, hogy az egész rendszerhez van hozzáférésük. Az Admin(ok)nak hozzá kell tudni férniük a felhasználók listájához, ahol mindent átváltoztathatnak egy felhasználó profilján. Tudniuk kell a felhasználók jogosultságait, szerepkörét, jelszavát, és felhasználónevét módosítani. Továbbá képesnek kell lenniük arra, hogy felhasználókat vegyenek fel rendszerbe és, hogy rakjanak le belőle. Fontos, hogy ők is képesek feladatok létrehozni, mint a tanárok. Képesek üzenetet küldeni az összes felhasználónak, valamint globális üzeneteket, amelyet mindenki megkap egyszerre. A Diák jegyeit csak ők tudják módosítani, miután a Tanár adott neki."*

7. Megfeleltetés, hogyan fedik le a használati esetek a követelményeket

-

8. Képernyő tervek

- A képernyő tervek mutatják meg, hogy mely funkciók kerülnek egymás mellé, melyik képernyőről mely képernyőre juthatunk.
- **Példa:**

9. Forgatókönyv

- A forgatókönyvek a rendszer egy-egy tipikus felhasználását mutatják be. A forgatókönyvnek általában van egy célja, például egy iktató rendszerben: Levél érkeztetése, címzett értesítése, dokumentumtárba helyezés. A forgatókönyv bemutatja, milyen funkciókat kell használni, milyen sorrendben a kívánt cél elérése érdekében. Ilyen értelem egy telepítési útmutatóhoz hasonlítanak.
- **Példa:** *“Szereplők: Futási időben három szereplő figyelhető meg. Az első szereplő maga a futó alkalmazás. (weben/androidon) Bejelentkezve kilehet választani a kívánt játékot. Megjelenik a timer a segítségek, és a játékos feladat. Ezzel van interakcióban a második szereplő, maga a felhasználó, aki kitölti a tesztet, úgy hogy az időn ne lépjen túl, és ha szüksége van akkor igénybe veheti a segítségek egyikét A harmadik szereplő egy web-service, ami a tesztekhez szükséges adatokat szolgáltatja az alkalmazásnak egy adatbázisból.”*

10. Funkció - követelmény megfeleltetés

-

11. Fogalomszótár

- A fogalomszótár a dokumentációban megemlített idegen esetleg nem egyértelmű jelentésű szavak / szakszavak pontos meghatározását írja le.
- **Példa:** “[web-service]:különböző programnyelveken írt és *különböző platformokon futó szoftveralkalmazások interneten keresztül történő adatcseréjére használt vebszolgáltatások.*
  [multiplatform]: több környezetben futtatható alkalmazás.
  [main menu]: A fő menü, amely a weboldal indulásakor megjelenik.

”

Rendszerterv

1. A rendszer célja

- Leírja hogy mit szeretne megoldani a rendszer.
- **Példa:** ”A rendszer célja, hogy a felhasználó játékos körülmények között tud tanulni, különböző feladatokat megoldani. A felhasználó pontszámokat kap arról, hogyan sikerült megoldania a feladatokat. Fontos, hogy a felhasználó könnyen el tudjon igazodni a felületeken ezért minimalista felhasználói felületet kap a program. A tanár szerepkörrel rendelkező felhasználók feltölthetnek feladatsorokat az adatbázisba. A rendszer használható Androidos eszközökön, alkalmazás formájában, valamint webes felületen is elérhető. A rendszer az adatokat egy Web Service segítségével kapja az adatbázisból. Mivel az alkalmazást csak webes felületen, és Android alkalmazásban szeretnénk elérhetővé tenni, nem célunk hogy más, például IOS operációs rendszerrel rendelkező eszközön fusson. A felhasználó a feladatsorok megoldása után pontszámokat kap. Teljesítményét a toplistán is megtekintheti.”

2. Projektterv

- Leírja a szerepköröket ,kik vannak a csapatban és min dolgoznak. Ide kerül az ütemterv és általában mérföldköveket tartalmaz.
- **Példa:** ”

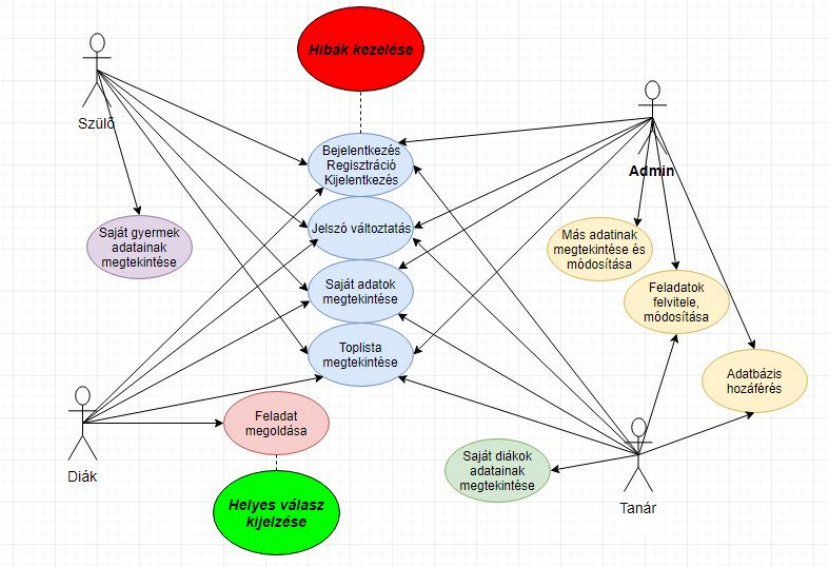
  Projektszerepkörök, felelőségek:
  Scrum master: Kiss Jenő
  Product owner: Nagy Béla
  Projektmunkások és felelőségek:
  Backend munkálatok: (csapat tagjai)
  Feladatuk az adatok tárolásához szükséges adatszerkezetek kialakítása, funkciók létrehozása, a különböző platformok kiszolgálása adatokkal.
  Ütemterv:

Funkció / Story	Feladat / Task	Prioritás	Becslés	Aktuális becslés	Eltelt idő	Hátralévő idő
Követelmény specifikáció		0	12	12	12	0
Funkcionális specifikáció		0	12	12	12	0
Rendszerterv		0	16	16	8	8
Adattárolás	Adatmodell megtervezése	0	4	4	4	0
	Adatbázis megvalósítása a szerveren	1	1	1	0	1
Login felület	Logó elkészítése	2	8	8	0	8

Mérföldkövek: Az adatmodell bemutatása megtörtént.”

3. Üzleti folyamatok modellje

- **Példa:**



4. Követelmények

- Leírja nagyvonalakban miket kell teljesítenie a programnak.
- **Példa:** “Funkcionális követelmények:

  - Felhasználók adatainak tárolása.
  - Felhasználók csoportokba szervezése.
  - Androidon és webes környezeten való működés.

Nem funkcionális követelmények:

  - A felhasználók nem juthatnak hozzá más felhasználók személyes adataihoz a nevükön és azonosítóikon kívül.

Törvényi előírások, szabványok:

  - GDPR-nek való megfelelés.”

5. Funkcionális terv

- Leírja a felhasználói szerepköröket, és hogy milyen feladatokat tudnak csinálni.
- **Példa:”**
  Rendszerszereplők:
  Admin
  Diák

Rendszerhasználati esetek és lefutásaik:

ADMIN:

- Beléphet bármilyen szereplőként teljes hozzáférése van a rendszerhez
- A felhasználói adatokat látják, változtathatják
- Felhasználó hozzáadására, törlésére van lehetőségük
- Feladatlétrehozás mint a Tanárok
- Diákok jegymódosítása
- Üzenetküldés bárkinek vagy globálisan
- Felhasználói adatok módosítása
- Tesztek létrehozása, törlése, módosítása
- Kvízek létrehozása, törlése, módosítása

DIÁK:

- Képes kvízt kitölteni, aminek végén pontot szerez
- Képes üzenetet küldeni más diákoknak, tanároknak vagy diákoknak
- El tudja érni az órarendjét
- Teszt felület elérése, ami egy kvízhez hasonló felület ahol eredményjegyet szerezhet a diák.

Menü-hierarchiák:

- BEJELENTKEZÉS
  - Bejelentkezés
  - Regisztráció
  - Help
- MAIN MENÜ
  - Kvíz
  - Feladatlétrehozás (Tanároknak)
  - Órarend megtekintés (Diákoknak)
  - E-naplo elérése
  - Személyes adatok
  - Toplist
  - Kijelentkezés”

6. Fizikai környezet

- Itt kell leírni milyen platformra készül a szoftver ,mik vannak engedélyezve és milyen fejlesztői eszközöket / programokat használunk fejlesztés közben.
- **Példa:”**

  - Az alkalmazás Android és web platformra, hordozható eszközökre(okostelefonok,táblagépek) készül.
  - Van tűzfal a hálózaton és minden portot is engedélyez.
  - Nincsenek megvásárolt komponenseink.
  - Fejlesztői eszközök:
    - Notepad++
    - Gradle

- MySQL Workbench
- Lumen Framework”

7. Absztrakt domain modell

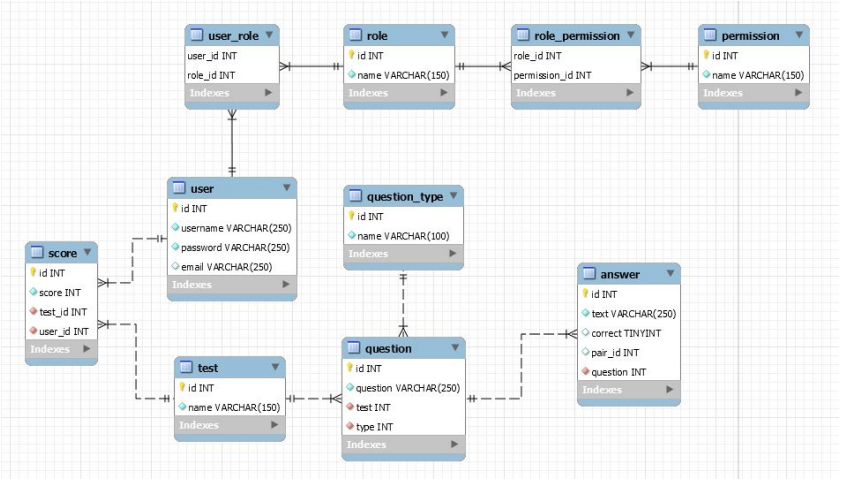
-

8. Architekturális terv

- **Példa:** “Backend:
  A rendszerhez szükség van egy adatbázis szerverre, ebben az esetben MySql-t használunk. A kliens oldali programokat egy php alapú REST api szolgálja ki, ez csatlakozik az adatbázis szerverhez. A kliensekkel JSON objektumokkal kommunikál.
  Web Kliens:
  A web alkalmazás Angular JS keretrendszer használatával készül el.A rest api-hoz a user belépését követően egyedi api-key segítségével lehet hozzáférni, ez biztosítja, hogy illetéktelen felhasználók ne módosíthassák az adatokat. ”

9. Adatbázis terv

- **Példa:**



10. Implementációs terv

- Leírja milyen technológiákat használunk hogyan és miért.
- **Példa:**”
  Web:
  A Webes felület főként HTML, CSS, és Javascript nyelven fog készülni. Ezeket a technológiákat amennyire csak lehet külön fájlokba írva készítjük, és úgy fogjuk egymáshoz csatolni a jobb átláthatóság, könnyebb változtathatóság, és könnyebb bővítés érdekében. Képes lesz felhasználni a Backend részen futó REST szolgáltatás metódusait, ezáltal tud felvinni és lekérdezni adatokat az adatbázisból. Az eltelt időt a kliens fogja számolni a feladatoknál, hogy ne legyenek eltérések.”

11. Teszterv

- Leírja a tesztelés folyamatát mikor milyen tesztek lesznek elvégezve.
- **Példa:"** A tesztelések célja a rendszer és komponensei funkcionalitásának teljes vizsgálata, ellenőrzése, a rendszer által megvalósított üzleti szolgáltatások verifikálása.
  Tesztelési eljárások
  Unit teszt:
  Ahol csak lehetséges, szükséges már a fejlesztési idő alatt is tesztelni, hogy a metódusok megfelelően működnek-e.
  Ezért a metódusok megfelelő működésének biztosítására mindegyikhez írni kell Unit teszteket, a minél nagyobb kódlefedettséget szem előtt tartva. A metódusok akkor vannak kész, ha a tesztesetek hiba nélkül lefutnak az egyes metódusokon.

  Alfa teszt:

  A teszt elsődleges célja: az eddig meglévő funkcióknak a különböző böngészőkkel, és androidokkal való kompatibilitásának tesztelése. A tesztet a fejlesztők végzik.
  Az eljárás sikeres, ha különböző böngészőkben és különböző androidokon is megfelelően működnek a különböző funkciók. A teszt időtartama egy hét.

  Beta teszt:

  Ezt a tesztet nem a fejlesztők végzik.
  Tesztelendő böngészők: Opera, Firefox, Google Chrome, Safari
  Tesztelendő android rendszerek:6.0.0(minimum), vagy újabbak
  Tesztelendő kijelző méretek: 1280x720 (minimum), 1366x768, 1920x1080

  A teszt időtartama egy hét.
  A tesztelés alatt a tesztelő felhasználók visszajelzéseket küldhetnek a fejlesztőknek, probléma/hiba felmerülése esetén.
  Ha hiba lép fel, a fejlesztők kijavítják a lehető leghamarabb. Sok hiba esetén a tesztelés ideje elhúzódhat plusz egy héttel.

  Tesztelendő funkciók
  Backend Service

  Képesnek kell lennie csatlakozni webes, és androidos klienshez is.
  Képesnek kell lennie egy időben kiszolgálni több klienst is.
  Fel kell tudnia tölteni, és le kell tudnia kérdezni az adatbázis adatait.
  Képesnek kell lennie minden felületen elérhető funkciók biztosítására.

  Android
  Login felület:

A login/regisztrációs felület elrendezésének ellenőrzése: Elvárt működés: a funkcionális specifikációban szereplő képernyőtervnek megfelelően kell kinéznie, a képernyő méretétől függetlenül.
Regisztrációs felület:

A regisztrációs felületnek elérhetőnek kell lennie a program telepítés után a kezdőképernyőn a bejelentkezési lehetőség mellett. Amennyiben a felhasználó még nincs regisztrálva az itt található gombra kattintva kell átirányítani a regisztrációs felületre. Ezen felületen a megfelelő adatok megadása mellett a megerősítés gombra kattintva a felhasználó regisztrációjának a funkcionális specifikációban leírtak szerint végbe kell mennie, majd elérhetővé kell tenni a bejelentkezést a felhasználó számára. Hibás regisztrációs adatok megadásakor hibaüzenetet kell kapjon a felhasználó.**”**

12. Telepítési terv

- Leírja hogyan kell telepíteni a programot.
- **Példa: *“*Androidos alkalmazás**
    - Töltse le az alkalmazást a Google áruházból, adja meg a szükséges engedélyeket és telepítse a programot!
    - Amennyiben nem az áruházból kívánja telepíteni az alkalmazást, úgy engedélyezze készülékén az úgynevezett "Harmadik féltől származó tartalmakat" a beállításoknál!
    - Helyezze az ".apk" kiterjesztésű elemet a készülékére, majd futtassa azt!
- **Webes alkalmazás**
    - A szoftver webes felületéhez csak egy ajánlott böngésző telepítése szükséges (Google Chrome, Firefox, Opera, Safari), külön szoftver nem kell hozzá. A webszerverre közvetlenül az internetről kapcsolódnak rá a kliensek.

13. Karbantartási terv

- **Példa: *“***Az alkalmazás folyamatos üzemeltetése és karbantartása, mely magában foglalja a programhibák elhárítását, a belső igények változása miatti módosításokat, valamint a környezeti feltételek változása miatt megfogalmazott program-, illetve állomány módosítási igényeket. Ellenőrizni kell, hogy a jövőben kiadott Android verziókkal kompatibilis-e az alkalmazás. Idő elteltével új kategóriákat kell hozzáadni a játékhoz, hogy fent tartsuk az érdeklődési szintet.

Karbantartás

Corrective Maintenance: A felhasználók által felfedezett és "user reportban" elküldött hibák kijavítása.
Adaptive Maintenance: A program naprakészen tartása és finomhangolása.
Perfective Maintenance: A szoftver hosszútávú használata érdekében végzett módosítások, új funkciók, a szoftver teljesítményének és működési

megbízhatóságának javítása.
Preventive Maintenance: Olyan problémák elhárítása, amelyek még nem tűnnek fontosnak, de később komoly problémákat okozhatnak.