



Adatbázis alapú web alkalmazás fejlesztése

Készítette

Verebélyi Valentin

Programtervező Informatikus BSc

Témavezető

Dr. Tajti Tibor Gábor

Egyetemi Docens

EGER, 2024

Tartalomjegyzék

Bevezetés	3
1. Tervezés	4
1.1. A tervezéshez alkalmazott eszközök és technológiák	4
1.1.1. PlantUML	4
1.1.2. Dbdiagram	4
2. Fejlesztés	7
2.1. A fejlesztéshez alkalmazott eszközök és technológiák	7
2.1.1. Laravel	7
2.1.2. PHP	7
2.1.3. MySQL	8
2.1.4. PhpMyAdmin	8
2.1.5. XAMMP	8
2.1.6. Visual Studio Code	8
2.1.7. GitHub	8
2.1.8. GitHub Desktop	8
3. Tesztelés	10
3.1. A teszteléshez alkalmazott eszközök és technológiák	10
3.1.1. Cypress	10
4. Felhasználói dokumentáció	11
Összegzés	12
Irodalomjegyzék	13

Bevezetés

1. fejezet

Tervezés

1.1. A tervezéshez alkalmazott eszközök és technológiák

1.1.1. PlantUML

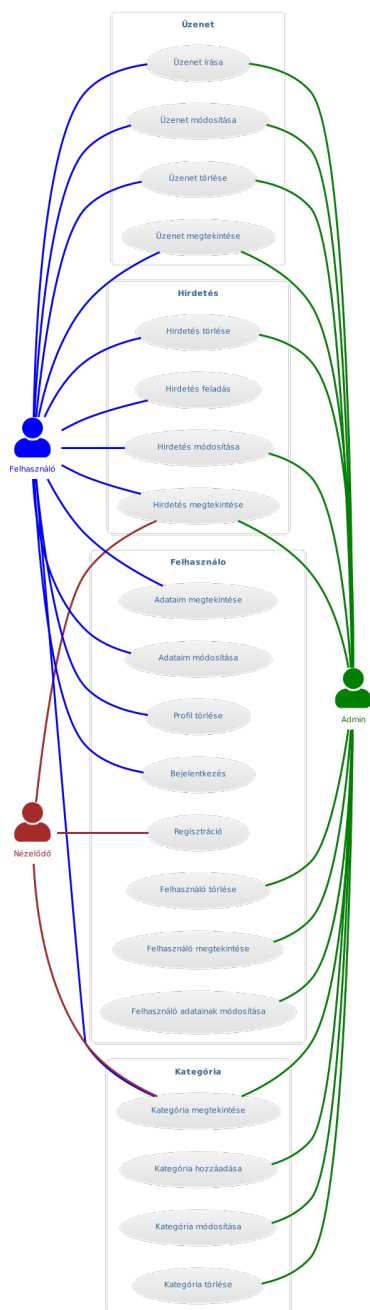
Az UML a Unified Modeling Language rövidítése, azaz Egységes Modellezési Nyelv. Az UML arra jó, hogy szoftvert tervezzünk vele, a gondolataimat letudom vele rajzolni, skiccelni. Ezenkívül a kiforrott megoldások dokumentálására is használható. A PlantUML egy komponens, aminek a segítségével gyorsan tudunk létrehozni szekvencia-, használati eset- és osztálydiagramokat. Ezenkívül még rengetegféle diagramot tudunk készíteni. [5]

A következő képen látható a plantUML-ben készített ábra az alkalmazásomhoz:

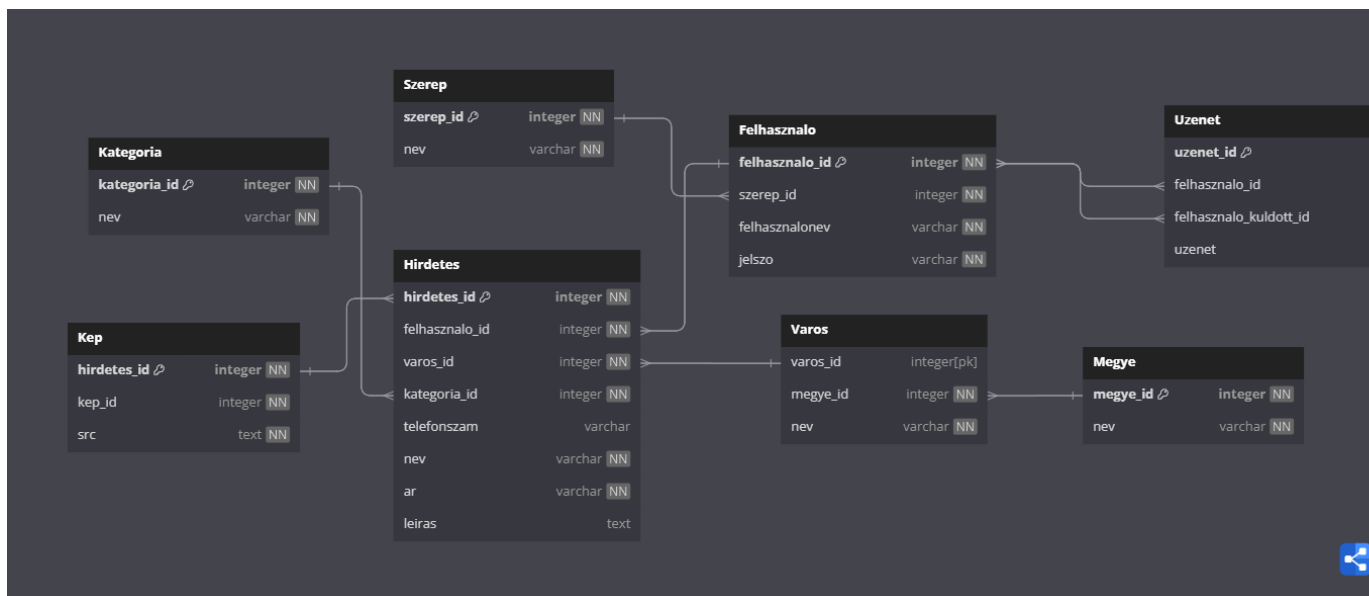
1.1.2. Dbdiagram

A dbdiagram segítségével van lehetőség arra, hogy egy alkalmazás adatbázisának a sémáját és struktúráját megtervezzük. A dbdiagram alap változata ingyenesen használható mindenki számára.

A következő képen látható a dbdiagram-ban készített ábra az alkalmazásomhoz:



1.1. ábra. Használati eset (Saját készítés)



1.2. ábra. Adatbázisterv (Saját készítés)

2. fejezet

Fejlesztés

2.1. A fejlesztéshez alkalmazott eszközök és technológiák

A következő alszakaszokban a fejlesztéshez használt eszközökről és technológiákról lesz majd szó.

2.1.1. Laravel

A Laravel egy nyílt forráskódú, PHP webkeretrendszer, ami MVC tervezési mintán alapszik. Az MVC a Model-View-Controller rövidítése.

1. Model (Modell): ez az adatokat kezelő réteg, az adatok tárolásáért és visszaolvasásáért felelős.
2. View (Nézet): ez a réteg felhasználói felületek megjelenítéséért illetékes.
3. Controller (Vezérlő): a felhasználói műveletek megfelelő kezeléséért ez a réteg a felelős.

Mivel három részre van bontva, ezért van ennek a tervezési mintának néhány előnye is. Ilyen például ha lecseréljük az egyik réteget, akkor a többihez nem kell már hozzányúlni, amivel pénzt és időt is lehet spórolni. Előnye még az is, hogy egy modellnek lehet több nézetes is. [1, 102-103. oldal]

2.1.2. PHP

A PHP egy szerveroldali szkriptnyelv, amelynek a segítségével dinamikus weboldalakat lehet vele létrehozni.

2.1.3. MySQL

A MySQL nem más, mint egy nyílt forráskódú adatbázis kezelő rendszer. A MySQL egy relációs adatbázis, ahol az adatokat különböző táblákban vannak eltárolva. A különböző táblákban szereplő mezők között lehetnek különféle kapcsolatok is. Ilyen lehet például az egy-az-egyhez kapcsolat vagy egy-a-többhöz kapcsolat. A MySQL adatbázisok szerverére jellemző, hogy gyorsak, megbízhatóak és skálázhatóak. [2]

2.1.4. PhpMyAdmin

A phpMyAdmin egy ingyenes szoftver, ami arra szolgál, hogy kezelje a MySQL-nek az adminisztrációját weben keresztül. A phpMyAdmin segítségével elvégezhetők a legtöbb adminisztratív feladatok, ideértve az adatbázis létrehozását, lekérdezések futtatását és felhasználói fiókok hozzáadását. [3]

2.1.5. XAMMP

Az XAMMP nem más, mint egy PHP fejlesztői környezet. Erről még azt érdemes tudni, hogy ingyenesen használható és rendkívül egyszerű telepítése, illetve a használata.

2.1.6. Visual Studio Code

Ez egy IDE (integrált fejlesztői környezet), ami ingyenes használható és az egyik legelterjedtebb és legnépszerűbb a fejlesztők körében. Fontosnak tartom megemlíteni, hogy a Visual Studio Code-ban van lehetőség különféle kiegészítők (extension) letöltésére is, például van lehetőség a python programozási nyelv vagy egy programozási nyelvhez tartozó szintaxis kiemelő letöltésére, illetve ezek használatára.

2.1.7. GitHub

A GitHub-ról azt érdemes tudni, hogy ez egy verziókövető rendszer. Ezek a rendszerek képesek állományok tartalmi változásait követni, azt is képesek megmondani, hogy ki és mikor módosította azokat, valamint van lehetőség arra is, hogy korábbi állapotokat is képes előállítani. A main branch-be feleltethető meg a fő ágnak, amiből van lehetőség elágazások (branch) is létrehozni. Az elágazásokat arra valóak, hogy a fejlesztési funkciókat elkülönítsük. Még arra is használhatjuk, hogy kísérletezzünk vagy akár hibajavítások elvégzésére is lehet használni.

2.1.8. GitHub Desktop

A GitHub Desktop egy ingyenesen használható alkalmazás, aminek a segítségével tudunk dolgozni a GitHub-on vagy Git-tárhely-szolgáltatásokon tárolt fájlokkal. Én ezt

az eszközt azért szeretem használni, mivel megkönnyíti és felgyorsítja számomra a munkavégzés, illetve azért is, mert ennek a használatához nem kell a terminálban beírni a Git-hez tartozó parancsokat, hanem egy-két kattintás segítségével elvégezhetek egy konkrét parancsot. [4]

3. fejezet

Tesztelés

A tesztelésre azért van szükség, hogy az alkalmazásomban megtaláljam az esetleges hibákat, amiket kijavítva növelhetem a szoftverem minőségét és megbízhatóságát. Abban sajnos nem lehetek biztos, hogy a tesztelés elvégzése után nem lesznek már hibák. A tesztelés során kettőféle tesztelési technikát alkalmaztam. Ezek pedig a következők:

1. Fehérdobozos (white-box): a forráskód alapján íródnak a tesztesetek.
2. Szürkedofozos (grey-box): amikor a forráskódnak csak egy rész ismert és csak ez alapján íródnak a tesztesetek.

Azonban van egy harmadik féle is, amit nem alkalmaztam. Az nem más, mint a fekete-dofozos (black-box), amikor a tesztesetek a specifikáció alapján íródnak. [1, 26-29. oldal]

3.1. A teszteléshez alkalmazott eszközök és technológiák

3.1.1. Cypress

A Cypress egy NodeJS-ben írt front end tesztelési keretrendszer. Ahhoz, hogy tudjuk futtatni a Cypress-t, ahhoz előtte telepíteni kell a NodeJS-t. A Cypress segítségével tudunk készíteni végponttól végpontig tartó-, komponens-, integrációs- valamint a unit-teszteket is. Ennek a tesztelési keretrendszer segítségével bármit lehet tesztelni ami a böngészőben fut. [6]

4. fejezet

Felhasználói dokumentáció

Összegzés

Irodalomjegyzék

- [1] KUSPER GÁBOR: *Programozási technológiák*, Eger, 2015.
- [2] ORACLE: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>, 2024-es verzió.
- [3] PHPMYADMIN: <https://www.phpmyadmin.net/>, 2024-es verzió.
- [4] GITHUB DESKTOP: <https://docs.github.com/en/desktop/overview/about-github-desktop>. 2024-es verzió.
- [5] PLANTUML: *PlantUML Language Reference Guide*, <https://plantuml.com/guide>, 2023. novemberi verzió.
- [6] CYPRESS: <https://docs.cypress.io/guides/overview/why-cypress>, 2024-es verzió.