PocketPharmacy

1. Projektterv

1.1 A választott feladat megnevezése és rövid leírása

Projektünkben egy otthoni gyógyszernyilvántartó rendszerhez készítünk alkalmazást. Az alkalmazás alapvető célja, hogy a felhasználó számára naprakész információval szolgáljon az otthonában fellelhető, általa szedett gyógyszerekről; azok kiszereléséről (30 tabletta/doboz), hatáserősségéről (pl. hány mg hatóanyag van az adott gyógyszerben), adagolásáról (napi 2x1 tabletta / heti 3x5ml), lejárati dátumáról, hogy az adott gyógyszer mire használható, illetve, mikor nem szedhető. Emellett azt is jelzi, hogy az adott gyógyszerből mikor kell újat vásárolni.
A felhasználónak lehetősége lesz az egyes gyógyszereket azok attribútumaival felvinni a rendszerbe, melyeket táblázatos formában meg is tud tekinteni. Emellett a felvitt gyógyszerek módosítására és törlésére is van lehetőség. Az alkalmazás a megjelenítéskor jelzi, ha a felhasználó valamely gyógyszerének lejár a szavatossága, valamint, ha az adott gyógyszerből a rendelkezésre álló adag kevesebb, mint egy hétre elegendő.
A táblázatban szereplő gyógyszerekre kattintva az alkalmazás bemutatja annak részletes adatait.

1.2 A választott fejlesztési módszertan, indoklással

Projektünk mielőbbi elkészülése és ezzel egyidejűleg funkcionális használata érdekében Agilis fejlesztési szoftverfejlesztési módszert választottunk.
Agilis szoftverfejlesztésben megfogalmazott szempontok alapján (Fő alap feladatok felbontása és fontossági sorrendbe állítása → Főbb feladatok apró véges feladatokra bontása → Apróbb feladatok határidőre való elvégzése majd beszámoló, észrevételek jelzése → Működő szoftver → Visszajelzések alapján Karbantartás/Módosítás) tudtunk a leglendületesebben felépíteni, részfeladatokra szétbontani és elvégezni a projekt feladatait.

Csapatunk fontosnak tartotta a projekt folyamatáról egymás folytonos értesítését, átlátható tájékoztatását. Először megbeszéltük a program alapvető funkcióit, miket tudjon az applikáció. Ezután elkezdtük felépíteni a frontend, backend, illetve adatbázis alapjait, kommunikációs formáit.

Nem szerettünk volna tesztelőknek félkész ‚alpha' programot kiadni, szerettünk volna egy stabil alappal rögtön egy megfelelő működő programot csinálni. Ezzel rengeteg időt megspórolva. Ezért is nem jöhetett szóba a prototípus alapú szoftverfejlesztés vagy a komponens alapú és újra felhasználható rendszerfejlesztés, illetve az evolúciós szoftverfejlesztési modell.

A jól átláthatóság érdekében még a tervezés fázisban szerettünk volna minden apró részletet megtervezni, hogy a fejlesztés alatt minél kevesebb zsákutcába, problémába és összezavarodottságba kelljen belefutni, amik csak hátráltatták volna a fejlesztés folyamatát. Ezért kellett kizárnunk a spirál fejlesztést, a vízesés modellt.

1.3 Használni kívánt fejlesztőeszközök, indoklással

Projektünk három alapvető komponensből fog felépülni: egy adatbázis, valamint maga az alkalmazás két részre bontva: front-end és back-end.

Az adatbázis szabvány szerint MySQL lesz - pontosabban egy MariaDb fork, amely az ingyenes használható XAMPP programcsomagon belül elérhető. Továbbá az ebben a programcsomagban elérhető adatbázis-kezelő rendszert fogjuk használni a MySQL adatbázis menedzselésére, a phpMyAdmin-t. Legfőbb indoka a MySQL használatának az alkalmazásunkban, hogy ingyenesen elérhető, könnyen kezelhető az adatbázis-kezelő rendszerben, valamint tudásunk erre az adatbázis termékre terjed ki legfőképp.

Az alkalmazást több rétegben és cross-platformon képzeljük el, ez adta az alábbiakban taglalt technológiák nélkülözhetetlen alkalmazását is.

A rendszer alapvetően egy webalkalmazás formájában fog elkészülni, amely front-endjét egy Angular-ban írt kliens alkalmazás fogja adni. Manapság egy komolyabb rendszer elkészítéséhez nélkülözhetetlenek az előre kész keretrendszerek felhasználása. A Google által fejlesztett és karbantartott Angular keretrendszer egy tökéletes választás a front-end

keretrendszerek végtelen tárházából, amellyel szép és kifejező kódok elkészítésében nyújt segítséget, hogy a kliens oldali technológiák (HTML, CSS, JavaScript) tökéletesen együttműködjenek. A webalkalmazás fejlesztéséhez Visual Studio Code fejlesztőkörnyezetet alkalmazunk.

Projektünk back-endjét egy C# nyelven fejlesztett Model-View-Controller tervezési architektúrán alapuló “RESTful API” fogja adni. A fejlesztése Visual Studio 2019 Community verziójú fejlesztőkörnyezetet fogunk alkalmazni - segítséget pedig a benne előre kész ASP.NET Core Web API project template fog nyújtani. Az ötlet, hogy a backend REST API formát öltsön, annak eredménye, hogy a korábban már taglalt rétegelt fejlesztést megvalósíthassuk. Ennek lényege, hogy a front-end és a back-end hálózaton képes egymással kommunikálni, valamint két eltérő erőforrással rendelkező szerverre, fizikailag egymástól függetlenül elhelyezhető - ezáltal a rendszer erőforrások tekintetében teljes mértékben skálázhatóvá válik. Mivel a back-end .NET Core alapokon készül el, bármilyen operációs rendszerrel rendelkező szerverre telepíthető - cross-platform jellegéből adódóan nincs az a gát, hogy csak Windows-t futtató szerverre telepíthető. Maga a REST API lényege, hogy egységes adattovábbító szabványban küldjön a kliensnek adatokat az adatbázisból - jelen esetben JSON formátumban -, amelyet bármely kliens alkalmazás fel tud dolgozni hálózati kapcsolat felhasználása során (asztali alkalmazás, mobilalkalmazás, webalkalmazás). Tovább gondolva a projektet, egy későbbi mobilalkalmazás elkészítéséhez csak a kliens-t kell kicserélni (pl. Android alkalmazás), a REST API-t érintetlenül hagyva képes lesz őt kiszolgálni adatokkal. Ezt kiegészítve, a külön szerverre, saját erőforrásokkal felvértezve egyidőben képes webalkalmazást és mobilalkalmazást is kiszolgálni egyidőben.

2. Feladatmátrix terve

Feladat	%	Bacsur Martin	Juhász Bence	László Noémi
feladatmátrix és projektterv (5%)	5%		5%	
rendszerspecifikáció (5%)	5%			5%
rendszerterv (40%)	40%			
* a rendszer elemzése, folyamatok feltárása			4%	
* adatbázisterv				12%
* interfésztervezés		12%		
* OOP terv (UML)			12%	
forráskód (35%)	35%			
* GUI		4%		
* programfunkciók				
** új felhasználó rögzítése		4%		
** új elem rögzítése			4%	
** elem módosítása		4%		
** elem törlése		4%		
** felhasználók karbantartása			2%	
** SQL parancsok				4%
** adatbázis használata				4%
** adatok vizualizációja		4%		
** kivételkezelése			1%	
tesztelési terv és jegyzőkönyv, a projekt értékelése (5%)	5%			
* tesztelés			2%	
* projekt értékelés				3%
felhasználói dokumentáció (5%)	5%			
* szövegezés				2,50%
* képernyőképek				2,50%
prezentáció (5%)	5%		5%	
	100%	32%	35%	33%