

Dokumentáció

zárdolgozathoz, vizsgaprojekthez

Jegyzet

Amikor szoftverfejlesztőként akár a [zárdolgozat](#) keretében, akár már egy "éles" munka részeként dokumentációt kell írunk egy elkészült programunkhoz, gyakran ez komolyabb kihívásnak számít, mint magának a szoftvernek az elkészítése. Az alább olvasható ötletgyűjtemény e munka megkönnyítéséhez szeretne kiindulási alapot, inspirációt, kiegészítő gondolatokat kínálni. Első közelítésből megnézzük a lehetséges **főbb fejezeteket**, majd ezt követően áttekintjük a legfőbb **formai elvárásokat**, végül pedig kiemelünk olyan **tartalmi kérdéseket**, melyek programnyelvtől, és környezettől függően ugyan, de jó eséllyel helyet kaphatnak a dokumentációban. Az oldal alján pedig található egy **Önellenőrző lista**, mellyel checkboxos felületen ellenőrizhető, hogy tartalmi és formai alapon milyen minőségű a dokumentációnk aktuális állapota. A felület százalékos értékelést és abból számolva körülbelüli osztályzatot is mutat.



Főbb fejezetek

Egy zárdolgozatként, vagy vizsgaprojektként készített programhoz a dokumentációban az alábbi fejezeteket érdemes kialakítani.

1. Bevezetés, a téma ismertetése, témaválasztás indoklása, szakmai célkitűzés

Itt azt célszerű leírni, hogy miről szól az elkészült program, milyen szakmai és/vagy hasznosulási céllal került megvalósításra, és mi indokolta a téma választását a készítő személy szemszögéből, élethelyzetéből nézve. Ez a bevezetés nagyjából **egy szűk oldalban** megvalósítható.

2. Fejlesztői dokumentáció

A fejlesztői dokumentáció a fejlesztés menetéről szól. Azért készül, hogy ha valaki szakmabeliként érdeklődik a program sajátosságai iránt, akkor beleláthasson a **technikai részletekbe**. Mivel szakmabeli olvassa, nem szükséges "szájbarágósnak" lenni, sőt kifejezetten az a cél, hogy szakmai nyelvezetet használjunk benne. Alapvetően **5 fő alfejezete** van:

- a **fejlesztőkörnyezet** (hardver és szoftverek) ismertetése, és a választás indoklása
- a kialakított **adatszerkezet** (pl. adatbázis, adatfájlok, főbb változók) részletes bemutatása
- a program tipikus, egyedi, különleges, vagy érdekesebb **algoritmusainak** bemutatása
- **tesztelés** leírása, tesztdokumentáció
- **fejlesztési lehetőségek** ismertetése

A **tesztdokumentáció** nem arról szól, hogy hol nem működik jól a program - hiszen a program mindenhol jól működik, különben nincs kész! -, hanem arról, hogy hogyan viselkedik különböző környezetekben, illetve nem optimális használat esetén. Ilyenformán érdemes benne leírni, hogy milyen hardverkörnyezetben (számítógépen,

vagy épp mobileszközön), valamint milyen operációs rendszeren, illetve pl. böngészőben, milyen képfelbontás mellett lett kipróbálva, és ott milyen sajátosságokat tapasztaltunk a teszt során. Továbbá ebben a részben kap helyet a helyes működés bizonyítása is, amikor kipróbáljuk a programunkat különböző rossz adatokkal, vagy éppen nagyon sok adattal, esetleg adatok nélkül, és bemutatjuk, hogy ezekre a helyzetekre hogyan - pl. milyen üzenetekkel - reagál a programunk. Ezeken túl szintén a tesztdokumentációban praktikus lehet szerepeltetni egy a belépéshez szükséges felhasználónév-jelszó párost, mely saját regisztráció nélkül szolgálhatja egy tesztfelhasználó számára a belépés és tesztelés lehetőségét.

Afejlesztési lehetőségek ismertetése arra biztosít lehetőséget, hogy elmondjuk, mi az, amilyen funkciókkal a program a későbbiekben tovább bővíthető, gazdagítható. Ezzel egyrészt jelezhetjük, hogy tudjuk, mit nem tud még a programunk, másrészt érzékeltethetjük, hogy a program nem öncélúan, kizárólag a vizsga miatt készült, hanem további terveink vannak vele. Ez a fejezet korrekt kifejtettség esetén jellemzően **16-20 oldal**, melyből a tesztdokumentáció 4-6 oldal.

3. Felhasználói dokumentáció

A felhasználói dokumentáció **a felhasználónak szól**. Annak a felhasználónak, aki azt sem tudja, hova csöppent, mit tart a kezében, mit kezdjen vele, és azt hogyan tegye. Éppen ezért ez a dokumentációrész **nagyon részletes**, nagyon "szájbarágós", és **nagyon a nulláról indul**. Nagyjából onnan például, hogy "ez egy számítógépes szoftver". Vagy hogy ez egy "közösségi weblap". És el kell benne mondani, milyen eszközön használható, milyen szoftver kell hozzá, hogyan kell letölteni, telepíteni, elindítani... stb.

E fejezet **6 részre bontható**:

- A program **céljának és lényegesebb funkcióinak** összefoglalása
- Szükséges **hardvereszközök és szoftverek** felsorolása
- **Telepítés és indítás** lépéseinek ismertetése
- A program **részletes bemutatása**
- Helytelen használatból adódó **hibajelzések** magyarázata
- **Információkérés** lehetőségeinek megadása

A **hibajelzések** magyarázata erős átfedést tartalmazhat a tesztdokumentáció helyes programműködést bizonyító részével. Az eltérés leginkább a megfogalmazásban, a nyelvezetben érhető tetten, ám a kiemelt képernyőrészletek, mint ábrák - melyek a különféle hibajelzéseket mutatják be - mindkét helyen felhasználhatóak. Itt lehet elmesélni, miként viselkedik a program pl. hiányosan kitöltött űrlapok, vagy éppen hibás regisztrációs, illetve bejelentkezési adatok megadása esetén, valamint találatot nem eredményező keresési feltételek mellett.

Az **információkérés** lehetőségének biztosítására érdemes a szerző által egy kifejezetten erre a célra létrehozott e-mail címet feltüntetni. Elegánsabb esetben az e-mail címen túl (és nem pedig helyette!) a program részeként egy űrlapot is létesíthetünk pl. *Kapcsolat* címszó alatt, amit szintén tüntessünk fel a dokumentációban.

Ez a fejezet, ha minden kívánt tartalmi elemet magába foglal, kevéssé úszható meg **10-12 oldal**nál kevesebből. Ellenkező esetben érdemes végiggondolni a fentieket, vajon mi maradt ki a leírásból.

4. Összefoglalás, köszönetnyilvánítás

A dokumentáció záró fejezete a némiképp **szubjektív gondolatoknak** adhat teret. Itt összefoglalható, hogy saját megítélésünk szerint mennyire sikerült megvalósítanunk az első fejezetben megfogalmazott szakmai célokat, **hogyan látjuk az elkészült munkánkat**, mint kész projektet. Írhatunk arról, mik jelentették a legnagyobb

kihívásokat, és **mik voltak azok a dolgok, amikből a legtöbbet tanultuk**. Vázolhatjuk a **programunk utóéletéről** alkotott elképzeléseinket, vagyis azt, hogy miként szeretnénk a folytatásban hasznosítani a művünket.

Továbbá itt lehet helye a köszönetnyilvánításnak is, melyben a megemléendő személyek tekintetében nemcsak **szakmai szempontok** alapján érdemes gondolkodni, hiszen egyrészt **családtagjaink, szeretteink** vélhetően toleranciával viseltettek irántunk, míg mi tanultunk, másrészt esetleg szüleink, eltartóink anyagi áldozatokat is hoztak szakmai fejlődésünk érdekében.

A fejezet jellemzően **1-2 oldal** terjedelmű szokott lenni.

5. Irodalomjegyzék

E részben kell felsorolni azokat a szakmai forrásokat, amiket a munkánk során felhasználtunk. Itt éppúgy érdemes megemlíteni néhány könyvet (szerző, cím, kiadó, kiadás éve), mint ahogyan weboldalakat is felsorolhatunk.

Szakmai könyveket akkor is érdemes megemlíteni az irodalomjegyzékben, ha csak érintőlegesen használtunk ilyeneket, mert emelik a dolgozatunk fényét. Ugyanakkor csak olyan könyvet jelöljünk meg, amelyről szükség szerint valóban tudunk legalább egy-két mondatot mondani a programunk szóbeli bemutatása alkalmával.

Weboldalak esetén nem elegendő megadni az oldal fő címét, hanem teljes mélységű URL-eket kell megjelölni. Pl. helytelen az infojegyzet.hu egyszerű megnevezéssel történő említése. A helyes jelölés:

- Űrlap-elemek megvalósítása HTML-ben <http://infojegyzet.hu/webszerkesztes/html/urlapok/>
- PHP alapok - szerveroldali programozás <http://infojegyzet.hu/webszerkesztes/php/alapok/>
- Dinamikus képgaléria <http://infojegyzet.hu/webszerkesztes/php/kepgaleria/>
- Záródolgozat témák, szempontok, ötletek <http://infojegyzet.hu/webszerkesztes/zarodolgozat/>

Az irodalomjegyzék általában rá szokott férni **1 oldalra**.

Formai elvárások

A dokumentáció formai elvárásainak tekintetében a következő irányelveket célszerű követni.

- A dokumentáció szövegtörzsének **betűmérete 12 képpont**.
- A dokumentáció szövegtörzse **másfeles sortávú**.
- A dokumentáció többsoros **bekezdései sorkizárt formázásúak**.
- A dokumentáció elején **tartalomjegyzék** található, mely a szövegszerkesztő tartalomjegyzék-készítő funkciójával készült, a szövegben alkalmazott **címsorok alapján**.
- A címsorok kialakítása során a fejezetek és alfejezetek **hierarchikus átláthatósága** érdekében célszerű **többszintű sorszámozást** alkalmazni.
- Az előfejlben, vagy az előlábban szerepelnie kell az **aktuális oldalszámnak**.
- Az előfej, illetve az előláb elegánsabb kialakítás esetén **vízszintes határoló vonalat** is tartalmazhat, és az előfejlben szerepeltethető a záródolgozat címe, vagy még igényesebb megoldásként **az aktuális fejezet címe**.
- Nyomtatáskor **egyoldalas nyomtatást** kell alkalmazni: ne nyomtassunk a lapok mindkét oldalára. Az előfej/előláb kialakításánál ezt vegyük figyelembe.
- A dokumentáció **minden fő fejezetét kezdjük új lapon**.

- Az elkészült dokumentációt olvassassuk át egy **jó helyesírással** rendelkező ismerősünkkel. Nagyon illúzióromboló, amikor egy dokumentáció tele van elgépelésekkel, mondatbéli vesszőhibákkal, vagy éppen következetesen helytelenül egybe-, illetve különírt igeekötőkkel és igékkel.
- A kész dokumentációt mentsük el **PDF formátumban** is. A PDF állománynak a program beadására szolgáló **adathordozón megtalálhatónak** kell lennie.

Tartalmi kérdések

A **fejlesztői dokumentációban** gyakran találhatóak nagyon jellemzően hiányos, vagy teljesen kimaradó elemek. Ilyenek lehetnek az alábbi példákban szereplők.

- Bármilyen feladat elkészítését is választottuk vizsgamunkaként, jó eséllyel létezik már hasonló program, weboldal, alkalmazás. Nevezzük meg ezeket, és indokoljuk meg, hogy az általunk készített szoftver miben más, mint a felsoroltak, vagyis **miért volt szükség az elkészítésére**.
- Bármilyen feladat elkészítését is választottuk vizsgamunkaként, egészen biztos, hogy azt többféle környezetben is meg lehetett volna valósítani. Éppen ezért meg kell indokolni, hogy **miért azt a programnyelvet, és miért azt a fejlesztőkörnyezetet választottuk**, amit.
- Az adatbázis ismertetésénél nem elegendő felsorolni az egyes táblákat. Részletesen le kell írni a **táblák** szerepét, és a bennük lévő **mezőket** az ő típusuk és hosszuk feltüntetésével. Szükség esetén indokolni kell döntéseinket. Továbbá célszerű készíteni az adatbázisról egy összefoglaló ábrát, melyben jelezzük a **táblák közötti kapcsolatokat**. Ez utóbbi ábra elkészítéséhez jelentős segítséget nyújthat a dbdesigner.net interaktív tervezőfelülete.
- Szükséges leírni, hogy milyen **jellemző, érdekesebb, egyedibb algoritmusok** készültek a program során. Egy ötlet példaként: Hogyan ellenőrzi a program, hogy a regisztrációkor megadott e-mail cím valid, és nincs ezzel a címmel más még nem regisztrált? Ilyen érdekesebb algoritmusból érdemes **4-5 darabot** kiválasztani, és bemutatni.
- Aki a programot fejlesztőként tesztelni fogja, nem biztos, hogy regisztrálni is akar. Biztosítsunk számára **tesztelői hozzáférést**. Ez lehet akár egy önálló oldal, mely megkerüli a regisztrációt, vagy pedig lehet egy kifejezetten erre a célra létrehozott speciális tesztelői felhasználónév-jelszó páros. Írjuk le ezt az opciót a fejlesztői dokumentációban.

Végül nézzünk meg néhány egyedi, de mégis tipikus, tartalomra vonatkozó lehetséges gondolatot a **felhasználói dokumentáció** kapcsán, mert általában itt szokott gyakori kérdés lenni, hogy *"Mit írjak még bele?"*

- Nagyon tipikus hiba, amikor a felhasználói dokumentáció így, vagy hasonlóképpen kezdődik:
- *"Az alábbi képen látható a program kezdőképernyője."*
- Mert hát ez rendben, de **hogyan és miért juthat el a kezdőképernyőig** a felhasználó? Miért jó neki, ha használja a programot? Ehhez kell egy számítógép? Vagy egy mobil eszköz? Milyen? Mit kell rajta elindítani? Mit kell beírni? Hova kell beírni?
- Szükséges hangsúlyosan tisztázni, hogy a programnak mik azok a funkciói, amiket **regisztráció nélkül** is használni lehet, és mik azok a lehetőségek, amik csak **regisztrációval** érhetőek el. Ennek hiányában a felhasználó nem fogja tudni, hogy miért érdemes neki regisztrálnia.
- Szintén tipikus hiba, ha a regisztrációt, illetve a belépést megvalósító felület ismertetése kapcsán nem kerül tisztázásra, hogy **milyen szabályoknak megfelelő felhasználónevet, valamint jelszót**

vár el a program. Hány karakterből kell állniuk minimum, illetve maximum? Milyen karakterek szerepelhetnek bennük? Alkalmazhatóak-e ékezetes betűk? Mi a helyzet a kisbetűkkel, és a nagybetűkkel?

- Egy képfeltöltő felület esetén mindig célszerű tisztázni a **feltöltendő képpel kapcsolatos elvárásokat**. Milyen formátumú kép tölthető fel? Mekkora lehet a kép legkisebb, illetve legnagyobb mérete pixelben? Van-e korlátozás a fájlméretre vonatkozóan? Lehetséges-e több képet is feltölteni? Feltöltés után átméretezésre kerülnek-e a képek? Milyen paraméterek alapján?
- **Keresőfelületek** bemutatásánál érdemes arról írni, hogy pl. lehet-e **összetett kulcsszavak** mentén keresni. Ha igen, akkor hogyan kell kötni egymáshoz a szavakat? Mi alapján, mely tartalmakban keres a rendszer? Cím? Leírás? Kommentek? Szükséges-e **ékezetes betűket** használni a keresőben, vagy anélkül is működik? Érzékeny-e a keresés a **kisbetűkre**, ill. **nagybetűkre**? Miként jelenik meg egy keresés eredménye? Olvasóként valószínűleg szívesen látna a felhasználó erről egy informatív képet.
- A fenti pont mintájára a felhasználók szeretnék látni a dokumentációban, hogy pl. egy-egy termékismertetés, vagy felhasználói profil, vagy receptoldalon egy recept, egy útvonaltervezőnél egy útvonal miként is néz ki a programban. Ne fukarkodjunk a különféle **felhasználói felületek képeinek** beszúrásával.
- Gondoljunk az **admin jogosultságú felhasználóra** is. Írjuk le, hogyan lehet létrehozni admin felhasználót, és ismertetjük az ő speciális lehetőségeit is a program használatának terén.

Természetesen a fenti ötletek alapján minden hasonló gondolatmenet helyet kaphat a vonatkozó dokumentációban. Újabb ötletek "beszerzése" pedig leghatékonyabban úgy valósítható meg, ha odaadjuk valamely ismerősünknek tesztelni a programot, és közben figyeljük, jegyzeteljük a cselekedeteit, és különösen a felmerülő kérdéseit. Élünk a lehetőséggel: **teszteltessünk!**

Minták a dokumentációra



Saját dokumentációjának készítése során beletekinthet már korábban elkészült munkák tartalmába is. Ennek során azonban fontos szem előtt tartani, hogy ezek a korábbi munkák sem tökéletesek, jószerével mindegyik

rendelkezik kevesebb, vagy több hiányossággal, ezért saját munkájának befejezéseként használja az alább önellenőrző listát!

Az alábbiakban e dokumentációra tekinthetőek meg példák a korábban elkészült munkákból.

- | | |
|---|---------------------------|
| <input type="checkbox"/> DanForum | Android alkalmazás |
| <input type="checkbox"/> Ékszerészek | PHP-MySQL program |
| <input type="checkbox"/> GameCreator | PHP-MySQL program |
| <input type="checkbox"/> Használt Babaholmi | PHP-MySQL program |
| <input type="checkbox"/> Játékbazár | PHP-MySQL program |
| <input type="checkbox"/> Kutyakozmetika | PHP-MySQL program |
| <input type="checkbox"/> MyCake | PHP-MySQL program |
| <input type="checkbox"/> Notam | Python-PostgreSQL program |

- | | |
|--|--------------------------|
| <input type="checkbox"/> OkosOtthon | Mikrovezérlő + C program |
| <input type="checkbox"/> Reptéri transzfer | PHP-MySQL program |
| <input type="checkbox"/> ThesisQuest | Python program |

Önellenőrző lista

Az alábbi összefoglaló **39 szempontja** alapján ellenőrizheti, mennyire precíz a dokumentációja. Jelölje meg azokat a négyzeteket, amelyeknek a tartalma saját megítélése szerint rendben van, majd a lista alatt tekintse meg a dokumentáció ezek alapján történő értékelését.



Bevezetés

1. Adtam címet a záródolgozatomnak.
2. Ismertettem a záródolgozatom témáját.
3. Megindokoltam a témaválasztásomat, leírtam az okokat.
4. Leírtam, hogy milyen funkciójú programot terveztem készíteni.
5. Megjelöltem a célközönséget, vagyis hogy kiknek szánom a programot.

Fejlesztői dokumentáció

6. Megindokoltam, miért volt szükség a program elkészítésére, és miben más, mint más hasonló létező program.
7. Ismertettem a fejlesztőkörnyezetem hardverét.
8. Ismertettem a fejlesztés során használt szoftvereket.
9. Megindokoltam, hogy miért azt a programnyelvet, és miért azt a fejlesztőkörnyezetet választottam, amit.
10. Részletesen írtam az adatszerkezetről. Adatbázis használata esetén bemutattam a táblákat, a táblák egyes mezőit, és készítettem ábrát a táblák közötti kapcsolatok szemléltetéséhez.
11. Bemutattam legalább 4 tipikus algoritmusomat.
12. A teszteldokumentációmban legalább 6 különböző körülményről, esetről, hibakezelésről írtam.
13. Leírtam, hogyan biztosítottam tesztelői hozzáférést (pl. adtam a belépéshez egy felhasználónév-jelszó párost).
14. Írtam fejlesztési legalább 2 fejlesztési lehetőséget.
15. A fejlesztői dokumentációm legalább 16 oldal.

Felhasználói dokumentáció

16. Néhány mondatban bemutattam, hogy a felhasználó milyen programot tart a kezében.
17. Leírtam, hogy a program használatához miféle hardver eszközre van szükség.
18. Leírtam, hogy a program használatához milyen szoftverekre van szükség.
19. Leírtam, hogyan lehet letölteni, telepíteni, elindítani a programot.
20. Részletesen ismertettem a program használatát.
21. Ismertettem, mely funkciók érhetőek el szabadon, és mihez szükséges regisztráció.
22. Minden a szövegbeviteli mező esetén leírtam, hogy ott mit vár el szoftver. Írtam a hosszról, megengedett karakterekről, ékezetes betűkről, kis- ill, nagybetűkről.
23. Szemléltetésként beszúrtam legalább 5 képernyőképet, vagy annak egy-egy részletét.
24. Bemutattam, milyen hibajelzéseket kaphat a felhasználó a tevékenysége során.
25. Külön ismertettem az admin felhasználó és az felület lehetőségeit.
26. Aadtam elérhetőséget információkérés lehetőségéhez.
27. A felhasználói dokumentációm legalább 10 oldal.

Összefoglalás

28. Leírtam, hogyan értékelem az elkészült munkámat.
29. Írtam róla, hogy hogyan látom a programom hasznosulását a továbbiakban (a vizsgától függetlenül).

Irodalomjegyzék

30. Az irodalomjegyzékben megneveztem legalább két általam is használt szakmai könyvet.
31. Az irodalomjegyzékben teljes mélységű URL-eket jelöltem meg az oldalak címével együtt, nem pedig csak domain neveket.

Formai szempontok

32. A dokumentációm másfeles sortávú, 12-es betűméretű.
33. A bekezdéseimre beállítottam a sorkizárt formátumot.
34. Tagolásként címsorokat alkalmaztam.
35. A fő fejezetek mindegyike új oldalon kezdődik.
36. A tartalomjegyzéket a címsorok alapján a szövegszerkesztőm készítette.
37. Az első oldal(ak) kivételével a folytatásban mindenhol látható az oldalszám az oldal tetején, vagy alján. Automatikus oldalszámozást használtam.
38. Munkámat helyesírás szempontjából ellenőriztettem valaki hozzáértővel. A szükséges korrekciókat elvégeztem.

39. Az elkészült dokumentációt PDF formátumba mentettem, és a PDF állományt felmásoltam adathordozóra a beadandó programom mellé.