Szakdolgozat

Formai követelmények:

A szakdolgozatot kemény kötésben kell leadni, 1 példányban.

Fedőlapjának színe fekete, aranyszínű feliratokkal.

Lap: A4-es méret, színe fehér

Betűméret: 12 pont

Sorok: sorkizárt igazítás, 1,5-es sortávolság

Margó:

- belső: 3,5 cm

- külső: 2,5 cm

- alsó: 2,5 cm

- felső: 2,5 cm

Oldalszám: az oldalszámozást a tartalomjegyzéktől kezdve az irodalomjegyzékkel bezárólag folyamatosan kell végezni. Jelölése arab számokkal történik (a tartalomjegyzék oldalainak az oldalszámát nem szokás feltüntetni).

## A dolgozat fő fejezetei (1. szintű címsorok) új oldalon kezdődjenek.

A szakdolgozat a hallgató önálló munkája, melyben be kell tartani a hivatkozások és idézések standard szabályait.

(lapszámozás, címsorok, tartalomjegyzék, stílusok, futó fejléc, ábraszámozás, stb.)

Belső fedőlap

Témabejelentő

Tartalomjegyzék

Tartalomjegyzék

[Oldalszám 1](#_Toc323154875)

[A dolgozat fő fejezetei (1. szintű címsorok) új oldalon kezdődjenek. 1](#_Toc323154876)

[Bevezetés 4](#_Toc323154877)

[Mit csinál ez a program 4](#_Toc323154878)

[Milyen környezetben 4](#_Toc323154879)

[Milyen eszközökkel 4](#_Toc323154880)

[Milyen hozzáállással készült 4](#_Toc323154881)

[Felhasználói dokumentáció 5](#_Toc323154882)

[Mire való a szoftver 5](#_Toc323154883)

[Célközönség 5](#_Toc323154884)

[Futtatási környezet 5](#_Toc323154885)

[Üzembe helyezés 5](#_Toc323154886)

[Használati útmutató 5](#_Toc323154887)

[Rendszerfunkciók 5](#_Toc323154888)

[Rendszerüzenetek 5](#_Toc323154889)

[Fejlesztői dokumentáció 6](#_Toc323154890)

[Megoldási terv 6](#_Toc323154891)

[Rendszer Architektúra 6](#_Toc323154892)

[Adatbázis 6](#_Toc323154893)

[Csomag, Modul és Osztályszerkezet 6](#_Toc323154894)

[Felhasználói Felület 6](#_Toc323154895)

[Megvalósítás 6](#_Toc323154896)

[#Ez mi a fene, cleancode-ról meséljek egy egész fejezetet? Oké... 6](#_Toc323154897)

[Tesztelés 7](#_Toc323154898)

[Junit 7](#_Toc323154899)

[MVP / Database interface / Sycron vs Asyncron hívások 7](#_Toc323154900)

[RDF Adatbázis mérete 7](#_Toc323154901)

[Eeh...? Eredményhelyesség - kihagyós 7](#_Toc323154902)

[Hatékonyság elemzés: GWT Speed Tracer 7](#_Toc323154903)

[Irodalomjegyzék 8](#_Toc323154904)

# Bevezetés

Mit csinál ez a program

(máv, utazás, érdekes helyek)

Milyen környezetben

(web, honlap, átlagembereknek)

Milyen eszközökkel

(GWT,web-service,külső REST api,RDF)

Milyen hozzáállással készült

(Clean code, Refactoring)

# Felhasználói dokumentáció

Mire való a szoftver

A program rövid ismertetése ~= bevezető? Szakmai ismertetés?

Célközönség

(kik, mikor, mire használhatják a programot)

## Futtatási környezet

A rendszer használatához szükséges minimális, illetve optimális HW/SW környezet  
Glassfishben a war beüzemelése

## Üzembe helyezés

Első üzembe helyezés leírása – ha van ilyen –, a program indítása (kivéve, ha nem egy önálló alkalmazásról, hanem egy meglévő rendszer új komponenséről van szó). Itt ellenőrizzük, hogy a telepítési útmutató megfelel-e a valóságos telepítési folyamatnak.

## Használati útmutató

Általános felhasználói tájékoztató (például a szokásostól eltérő képernyő-, billentyű-, illetve egérkezelés leírása, teendők hibaüzenetek esetén stb.).

## Rendszerfunkciók

A rendszer funkcióinak ismertetése. A feladat jellegéből fakadóan célszerű lehet ezt folyamatszerűen, képernyőképekkel alátámasztva bemutatni. A funkciókat ajánlatos a felhasználói szintek szerint csoportosítani. Itt vegyük figyelembe, hogy a leírás a fejlesztői dokumentációban meghatározott részfeladathoz illeszkedik-e, az ott meghatározott funkciókat/használati eseteket írja-e le?

## Rendszerüzenetek

A rendszer futás közbeni üzenetei (hibaüzenetek, figyelmeztető üzenetek, felszólító üzenetek stb.) és azok magyarázata – az esetleges üzemeltetési teendőkkel együtt. Itt vegyük figyelembe, hogy tartalmaz-e biztonsági, illetve hibaelhárítási előírásokat?

Egyéb, a szoftver használatához szükséges információk.

# Fejlesztői dokumentáció

## Megoldási terv

Ez a fejlesztői leírás része, a rendszerterv, amelyből az alkalmazás célja, felépítése és működése megérthető, ez alapján az alkalmazás forráskódja lényegében elkészíthető.

### Rendszer Architektúra

Rendszer architektúrájának leírását (alrendszerek, rétegek bemutatása, az alkalmazott szabványok, technológiák, fejlesztő módszerek megadása, felhasznált eszközök és kész komponensek definiálása). Az értékelésnél vegyük figyelembe, hogy mennyire válnak szét az alkalmazás rétegei (például felhasználói felület, logika, adatforrás)?

### Adatbázis

Az adatbázis – feltéve, hogy van – leírását. Érdemes egy áttekintő diagammal szemléltetni a táblákat és a köztük levő kapcsolatokat, majd külön táblázatokban megadni az egyes táblák mezőszerkezeti leírását, az esetleges tárolt eljárások, függvények, triggerek, stb leírását.

### Csomag, Modul és Osztályszerkezet

Modul és/vagy osztályszerkezet (fontosabb modulok és/vagy osztályok és azok metódusai, továbbá ezek kapcsolatának) leírását. Az egyes csomagok fő eljárásait illetve a fontos osztályok fő metódusait bemenő-, kimenőadat, tevékenység hármassal jellemezni kell.

### Felhasználói Felület

A felhasználói felület – feltéve, hogy van – tervét (a képernyő- és listaterveket, valamint a menütervet). Legyen egy áttekintő ábra, amely mutatja a képernyők (ablakok, weblapok) közti navigálási lehetőségeket, irányokat. Ki kell emelni a fontosabb felhasználói eseménykezeléseket.

## Megvalósítás

A fejlesztői leírásnak a megvalósításról szóló része bemutatja, hogy milyen döntéseket kellett hozni a terv megvalósítása során (adatábrázolás, felhasznált komponensek, kódban alkalmazott nyelvi elemek, stb). A dokumentáció ne tartalmazza a forrásprogramot (legfeljebb csak fontosnak ítélt részleteit), elég azt a mellékelt adathordozón elhelyezni. A megvalósítás a fentieken kívül tartalmazza a komponens tervet (az alkalmazás fizikai komponenseinek kapcsolatrendszerét) és azok telepítésének módját.

### #Ez mi a fene, cleancode-ról meséljek egy egész fejezetet? Oké...

A forráskód tartalma, szerkezete megfelel-e a tervnek?

 Mennyire ismeri a hallgató az adott fejlesztő eszközt (pl. korszerű, hatékony nyelvi elemek vannak-e túlsúlyban, vagy ehelyett bonyolult, nehézkes, körülményes és leginkább terjengős forráskódot eredményező nyelvi elemek jellemzik a kódot)? Indokoltak-e a választott nyelvi elemek használata?

 Milyen a forráskód külalakja, mennyire áttekinthető (strukturáltság, bekezdések, tagolások, kommentezés stb.)?

 Mennyire módosítható a kód. Alkalmazza-e a hallgató a kód-újrafelhasználás nyelvi eszközeit (függvények, származtatás, generikus elemek)?

 Törekszik-e a hatékony adatábrázolásra?

 Mennyire öndokumentáló a kód, vagyis a választott azonosítók (pl. változónevek) mennyire beszédesek, konvencionálisak, a megjegyzések mennyire segítik a kódértést?

 Tartalmazza a szükséges ellenőrzési, hibakezelési funkciókat, általában megoldott-e a kivételkezelés?

 Mennyire gazdálkodik jól az emberi és gépi erőforrásokkal, így például a felhasználó idejével és türelmével, a lemezkapacitással és a memóriakapacitással?

## Tesztelés

Ez is a fejlesztői leírás része, amelynek a tesztelési szempontokat kell bemutatnia, és a tesztelés során szerzett tapasztalatokat összegeznie valamint a szoftver skálázhatóságáról készített elemzést kell tartalmaznia.

Az értékelésnél vegyük figyelembe, hogy a dokumentáció:

### Junit

 Tartalmaz-e tesztelési terveket, teszteseteket (Ezeket csoportosíthatja rendszerteszt és modultesztek szerint illetve fekete és fehérdoboz megközelítéssel)?

### MVP / Database interface / Sycron vs Asyncron hívások

 Beszámol-e olyan tanulságokról, amelyek alapján meg kellett változtatni a korábbi implementációs döntéseket, esetleg a terv egyes elemeit (az ilyen tapasztalatok nem rontják a dolgozat értékét)?

### RDF Adatbázis mérete

 Tartalmazza-e nagy adattömeg melletti futtatások értékelését?

### Eeh...? Eredményhelyesség - kihagyós

 Elemzi-e a program által adott eredmény helyességét (különösen olyan optimalizációs feladatok esetén, ahol több helyes megoldást valamilyen célfüggvénnyel lehet rangsorolni)?

### Hatékonyság elemzés: GWT Speed Tracer

 Elemzi-e a program futásának hatékonyságát?

## 

# Irodalomjegyzék

GWT dokumentáció

Robert C. Martin – Clean Code

Martin Flower – Refactoring