# SZOFTVER KÖVETELMÉNY SPECIFIKÁCIÓ DOKUMENTUM

# BugTracker

2017.03.01

Tartalomjegyzék

1. Bevezetés [1](#__RefHeading___Toc476130127)

1.1 Célja [1](#__RefHeading___Toc476130128)

1.2 Dokumentum egyezmények [2](#__RefHeading___Toc476130129)

1.3 Érdekelt hallgatóság és olvasási javaslatok [2](#__RefHeading___Toc476130130)

1.4 Termék célja [2](#__RefHeading___Toc476130131)

1.5 Hivatkozások [2](#__RefHeading___Toc476130132)

2. Áttekintés [2](#__RefHeading___Toc476130133)

2.1 Termék perspektíva [2](#__RefHeading___Toc476130134)

2.2 Termék funkciói [2](#__RefHeading___Toc476130135)

2.3 Felhasználók és tulajdonságaik [3](#__RefHeading___Toc476130136)

2.4 Működési környezet [3](#__RefHeading___Toc476130137)

2.5 Tervezési és implementációs megszorítások [3](#__RefHeading___Toc476130138)

2.6 Felhasználói dokumentáció [3](#__RefHeading___Toc476130139)

2.7 Feltételezések és függőségek [3](#__RefHeading___Toc476130140)

3. Külső interfészek követelményei [4](#__RefHeading___Toc476130141)

3.1 User Interfaces [4](#__RefHeading___Toc476130142)

3.2 Hardware Interfaces [4](#__RefHeading___Toc476130143)

3.3 Software Interfaces [4](#__RefHeading___Toc476130144)

3.4 Communications Interfaces [4](#__RefHeading___Toc476130145)

4. Rendszer funkciói (Használati esetek) [4](#__RefHeading___Toc476130146)

4.1 Use Case 1 [5](#__RefHeading___Toc476130147)

4.1.1 Name: [5](#__RefHeading___Toc476130148)

4.1.2 Goal: [5](#__RefHeading___Toc476130149)

4.1.3 Input: [5](#__RefHeading___Toc476130150)

4.1.4 Output: [5](#__RefHeading___Toc476130151)

4.1.5 Main Scenario: [5](#__RefHeading___Toc476130152)

4.1.6 Pre-condition: [5](#__RefHeading___Toc476130153)

4.1.7 Steps: [5](#__RefHeading___Toc476130154)

4.1.8 Post-condition [5](#__RefHeading___Toc476130155)

4.1.9 Exceptional Scenario 1 [5](#__RefHeading___Toc476130156)

4.1.10 Example [5](#__RefHeading___Toc476130157)

4.2 Use Case 2 (and so on) [6](#__RefHeading___Toc476130158)

5. Nem funkcionális követelmények [6](#__RefHeading___Toc476130159)

5.1 Performance Requirements [6](#__RefHeading___Toc476130160)

5.2 Safety Requirements [6](#__RefHeading___Toc476130161)

5.3 Security Requirements [6](#__RefHeading___Toc476130162)

5.4 Software Quality Attributes [6](#__RefHeading___Toc476130163)

6. Más követelmények [6](#__RefHeading___Toc476130164)

# Áttekintés

## Dokumentum célja

Jelen specifikáció a BugTracker hibajegy rendszer tervezetét és szolgáltatásait mutatja be, elemzi a felhasználók által végezhető eseményeket és egy nagyobb rálátást bíztosít a projektben résztvevő emberek számára.

## Dokumentum fogalomtár

Fontos az itt tárgyalt fogalmak jelentését lefektetni a későbbi félreértések elkerülése végett.

Role, szerepkör – a jogosultságokat, privilégiumokat leíró fogalom. Belépést követően az egyes user-ek valamilyen szerepkörrel rendelkeznek, mely lehet: adminisztrátor, fejlesztő, felhasználó, jóváhagyó.

User – fontos megkülönböztetni a gyűjtőfogalmat és a szerepkörben foglalt felhasználót. A user magába foglalja az összes szerepkörben meghatározott felhasználót, míg a felhasználó az, aki bejelenti a hibákat.

Autentikáció – a rendszer funkcióit csak belépés után lehet elérni, melyeket az adott szerepkörhöz rendelten érhetnek el a user-ek.

Hibajegy – a projektekhez rendelt hibák a hibajegyek. A felhasználói szerepkörbe tartozó user-ek hozzák létre különböző sűrgősségi kategóriákba soroltan.

OS - operációs rendszer, amin a szoftver futni fog

Back end - a szoftver üzleti logikáját és az adatok perzisztálását teszi lehetővé ez a réteg

Front end - a szoftver felhasználóközeli részét valósítja meg az architektúra ezen rétege, ez a felhasználói felületet biztosító komponens

## Érdekelt hallgatóság

A dokumentum ajánlott mindenkinek, aki valamilyen módon kapcsolódik a projekthez, legyen az fejlesztő, megrendelő, felhasználó stb, mert segíthet a szoftver koncepciójának könnyebb megértésében, mely a fejlesztés és később majd a használat számos aspektusában kedvező lehet. A felhasználóknak elegendő lehet csak a 2. pontban tárgyalt **Használati eseteket** tanulmányozni, ha a rendszer működéséről szeretnének tájékoztatást kapni. Az egyéb olvasónak fontos lehet egészében áttekinteni a specifikációt.

## Termék célja

A BugTrecker rendszer az ügyfelek által jelentett, a fejlesztők által kezelt hibák, kérések állapotának követését biztosítja. Cél az, hogy a tömbösített bejelentések helyett egy rendszerezett, projektekre bontott, valamint sürgősségi besorolás szerint szétválogatott hibakezelő eszközt szolgáltasson, melyet az ergonómikus, könnyen kezelhető felhasználói interfész tesz egyszerűen használhatóvá. Bár több meglévő megvalósítása létezik az efféle szolgáltatásnak, ez az implementáció törekszik a minél általánosabb felület kialakítására, hogy a felhasználóknak egy széles spektrumú palettát nyújtson a hibák bejelentésére.

## Felhasználók jellemzői

A felhasználói szerepkör négy típusú user-ből tevődik össze. Ebben a részben ezt bővebben kifejtjük.

* Adminisztrátor

Az adminisztrátori szerepkör az a szerepkör, amely a legtöbb jogosultsággal rendelkezik. Egy adminisztrátor létrehozhat projekteket, felvehet user-eket – szerepkörbe besoroltan -, valamint ezeket a user-eket hozzárendelheti projektekhez, akár többhöz is. Természetesen erre nem csak a user felvételekor, hanem bármikor lehetősége van. Ugyanígy ezek ellenkezőjét is megteheti, azaz törölhet projektet, és eltávolíthat user-eket projektekből.

* Fejlesztő

A fejlesztők azok, akik az egyes hibajegyeket feldolgozzák és a hozzájuk rendelt feladatokat elvégzik. A fejlesztőknek lehetőségük van a hibajegyet létrehozó felhasználónak üzenetet küldeni a feladat pontosítása, tisztázása érdekében. Mikor egy fejlesztő foglalkozni kezd a hibajegyben leírt feladat elvégzésével, mindenekelőtt egy jóváhagyónak továbbítja a jegyet, aki eldönti, hogy a rendszer keretei közt megvalósítható-e a feladat. Ha minden adott, a fejlesztő elvégzi a munkát, majd a hibajegy tulajdonosához továbbítja a hibajegyet, hogy az hitelesítse, hogy a hibajegyben foglalt feladatot megfelelően elvégezték.

* Felhasználó

A felhasználók tekinthetők fogyasztóknak. Ők azok, akik az egyes szoftvereket használva hibákat találnak. Ezeket a hibákat a projektekhez rendelve feljegyezhetik, hogy a fejlesztők elvégezzék a megfelelő módosításokat a hiba elhárításának érdekében. A hibajegyek leadása után érkezhet kérdés a felhasználónak a feladattal kapcsolatban. Ezeket a hibajegyeket sűrgősségi kategóriákba besorolva hozhatják létre S1-S3 jelzésekkel, ahol S1 a legsűrgősebb. A felhasználónak hitelesítenie kell, hogy az általa feljegyzett hibát megfelelően elvégezték. Ezt akkor teheti meg, mikor a hibajegy a fejlesztőktől visszakerül hozzá megerősítésre.

* Jóváhagyó

Ebbe a szerepkörbe tartozó user-ek azok, akik jóváhagyják az egyes hibajegyekben tárgyalt hibák javítását, vagyis eldöntik, hogy a hiba a projekt keretei között van-e, vagy teljesen kívül esik azon. Ha elfogadják, az azt jelenti, hogy a fejlesztők elkezdhetnek dolgozni az adott feladaton, egyébként elutasított állapotba kerül majd a hibajegy.

## Tervezési és implementációs megszorítások

A megrendelő kérésére Linux operációs rendszeren fog a fejlesztés folyni, bár a további megkötések lehetővé teszik a multiplatform támogatását. Mivel az OS mellett mind a *front* és *back end* kötött, rendre: valamilyen JavaScript keretrendszer (a megrendelő javaslata az extjs), illetve egy Java nyelvű webszerver, így a portolhatóságot a Java Virtuális Gép biztosítja. Az adatok perzisztálását egy MySQL adatbázis fogja végezni.

## Feltételezések és függőségek

Feltételezhetjük, hogy a célgép Ubuntu 14, illetve attól frissebb operációs rendszerrel fog rendelkezni, emellett telepítve lesz minimum 1.7-es verziószámú Java Futtató Környezet. A webes kliens Google Chrome lesz, ezzel fog folyni a tesztelés és a fejlesztés egyaránt.

# Rendszer funkciói (Használati esetek)

## Használati eset 1

### Név: Bejelentkezés

### Aktor: Ügyfél / Fejlesztő / Adminisztrátor / Jóváhagyó

### Cél: A felhasználó jogosultságainak megfelelően tudja használni az alkalmazást.

### Bemenet: Bejelentkezési felület

### Kimenet: Bejelentkezett felhasználó

### Leírás: A felhasználó a bejelentkezési felületen beírja a felhasználó nevét és jelszavát. Sikeres esetben a felhasználó használhatja az rendszert jogosultságainak megfelelően.

### Kivételes eset: A bejelentkezés nem sikeres, a jelszó és felhasználónév nem talál.

## Használati eset 2

### Név: Új ügyfél, fejlesztő hozzáadása

### Aktor: Adminisztrátor

### Cél: Új felhasználó hozzáadása a rendszerhez.

### Bemenet: Bejelentkezett Adminisztrátor

### Kimenet: Az új felhasználó be tud jelentkezni a rendszerbe.

### Leírás: Az Adminsztrátor beírja a létrehozni kívánt felhasználó adatait a megfelelő űrlapra és elmenti azt.

### Kivételes eset: A felhasználó már létezik, ez esetben nem lehet újra létrehozni.

## Használati eset 3

### Név: Új projekt létrehozása

### Aktor: Adminisztrátor

### Cél: Az új project része a rendszernek.

### Bemenet: Bejelentkezett Adminisztrátor

### Kimenet: Az új project része a rendszernek

### Leírás: Az adminisztrátor a projekt létrehozó oldalon kitölti a projekt paramétereit és elmenti azt a rendszerbe.

### Kivételes eset: A projekt már létezik, ez esetben nem lehet újra létrehozni.

## Használati eset 4

### Név: Ügyfél / Fejlesztő / Jóváhagyó projekthez rendelése

### Aktor: Adminisztrátor

### Cél: Felhasználók hozzárendelése egy projekthez a megfelelő jogosultságokkal.

### Bemenet: Bejelentkezett Adminisztrátor, létező project és felhasználók

### Kimenet: A felhasználók megkapták a megfelelő jogosultságokat az adott projekthez.

### Leírás: Az adminisztrátor az adott projekthez navigál és hozzáadja a felhasználókat kiválasztva mindenik esetén a kívánt jogosultságot: Ügyfél, Fejlesztő, Jóváhagyó

### Kivételes eset: A projekt vagy a felhasználó nem létezik, ez esetben a művelet nem hajtható végre.

## Használati eset 5

### Név: Új hibajegy létrehozása

### Aktor: Ügyfél

### Cél: Új hibajegy létrejöjjön.

### Bemenet: Bejelentkezett ügyfél

### Kimenet: A hibajegy létezik a rendszerben új állapotban.

### Leírás: Az ügyfél létrehozza az adott hibajegyet a megfelelő leírással. Hozzárendel egy prioritást az S1, S2, S3 lehetőségek közül választva.

### Kivételes eset: A beírt névvel már létezik hibajegy.

## Használati eset 5

### Név: Új kérés létrehozása

### Aktor: Ügyfél

### Cél: Új kérés létrejöjjön.

### Bemenet: Bejelentkezett ügyfél

### Kimenet: A kérés létezik a rendszerben új állapotban.

### Leírás: Az ügyfél létrehozza az adott kérést a megfelelő leírással. Hozzárendel egy prioritást az S1, S2, S3 lehetőségek közül választva.

### Kivételes eset: A beírt névvel már létezik kérés.

## Használati eset 7

### Név: Hibajegyhez/Kéréshez fűzött kérdés megválaszolása

### Aktor: Ügyfél

### Cél: Várakozó állapotban levő, kérdéssel ellátott hibajegy/ kérés folyamatban van állapotba kerül.

### Bemenet: Kérdéssel ellátott várakozó állapotban levő hibajegy/kérés.

### Kimenet: A kérdéshez válasz van fűzve, az elem állapota folyamatban van.

### Leírás: Az ügyfél megnyitja avárakozó állapotban levő elemet és megadja a kerdesre a választ. Az állapotát átállítja folyamatban levőre.

### Kivételes eset: -

## Használati eset 8

### Név: Elkészült hibajegy/kérés újranyitása

### Aktor: Ügyfél

### Cél: Az elkészült hibajegy/kérés újra nyitott legyen.

### Bemenet: Elkészült hibajegy/kérés.

### Kimenet: Újranyitott hibajegy/kérés.

### Leírás: Az ügyfél az elkészült hibajegyet/kérést újranyitott állapotba rakja, opcionálisan megjegyzést főzve hozzá.

### Kivételes eset: -

## Használati eset 9

### Név: Elkészült hibajegy/kérés lezárása

### Aktor: Ügyfél

### Cél: Az elkészült hibajegy/kérés le legyen zárva.

### Bemenet: Elkészült hibajegy/kérés

### Kimenet: Lezárt hibajegy/kérés

### Leírás: Az ügyfél az elkészült hibajegyet/kérést lezárt állapotba teszi.

### Kivételes eset: -

## Használati eset 10

### Név: Elutasított kérés újranyitása

### Aktor: Ügyfél

### Cél: Elutasított kérés újranyitott űllapotba helyezése.

### Bemenet: Elutasított kérés

### Kimenet: Újranyitott kérés

### Leírás: Az ügyfél az elutasított kérést újranyitja opcionálisan megjegyzést fűzve hozzá.

### Kivételes eset: -

## Használati eset 11

### *Név:* Kérdéssel ellátott hibajegy/kérés függő állapotba tétele

### Aktor: Fejlesztő

### Cél: Ha fejlesztő számára nem világos minden részlet a probléma megoldásával kapcsolatban, kérdést fűzhet a hibajegyhez, az állapotát pedig függő-vé teheti.

### Bemenet: Egy kérdéssel ellátott hibajegy.

### Kimenet: A bemeneti hibajegy, melynek állapota függő.

### Leírás: A fejlesztőnek kérdése volt az ügyfél felé a hibajeggyel kapcsolatban. Kérdését a hibajegyhez fűzte, s a hibajegy állapotát “függő”-re állította. Ezzel a hibajegy visszakerül az ügyfélhez, hogy megválaszolhassa a felmerülő kérdést.

### Kivételes eset: Ha nincs kérdés a hibajegyhez, ne tudjuk függőben lévő állapotra állítani.

## Használati eset 12

### *Név:* Függő állapotban lévő hibajegy/kérés visszaállítása várakozó állapotba

### Aktor: Ügyfél

### Cél: A hibajeggyel kapcsolatos, fejlesztő részéről felmerülő kérdések megválaszolása.

### Bemenet: Kérdéssel ellátott, függő állapotban lévő hibajegy.

### Kimenet: A bemeneti hibajegy kérdései megvannak válaszolva, illetve a hibajegy állapota várakozó.

### Leírás: A fejlesztő felől kérdés érkezett a hibajeggyel kapcsolatban az ügyfélhez. Az ügyfél megválaszolja a kérdést, a hibajegy állapotát pedig várakozóra állítja, ezzel visszakerül a fejlesztőkhöz a hibajegy.

### Kivételes eset: Az ügyfél ne tudjon megválaszolatlan kérdéssel ellátott hibajegyet visszadobni a fejlesztőkhöz.

## Használati eset 13

### *Név:* Új hibajegy állapotának váltása “folyamatban lévő” állapotra

### Aktor: Fejlesztő

### Cél: A fejlesztő jelzi hogy el szeretné kezdeni a hibajegyben szereplő probléma megoldását.

### Bemenet: Egy új hibajegy.

### Kimenet: A bemeneti hibajegy, melynek állapota “folyamatban lévő”.

### Leírás: A fejlesztő kiválasztotta az új hibajegyek közül az aktuális jegyet, és jelzi, hogy el szeretné kezdeni a javítását. A munka megkezdéséhez szükség van egy jóváhagyó szerepkörű felhasználó visszaigazolására.

### Kivételes eset: -

## Használati eset 14

### *Név:* Folyamatban lévő hibajegy/kérés “elkészült” állapotra váltása

### Aktor: Fejlesztő

### Cél: Jelezni a jegy feladójának, hogy a fejlesztő úgy gondolja, megoldotta a problémát.

### Bemenet: Egy hibajegy, melynek állapota “folyamatban lévő”.

### Kimenet: A bemenetként kapott hibajegy állapota “elkészült”-re változik.

### Leírás: A fejlesztő úgy gondolja, hogy megoldotta a problémát, amit a hibejegyben jelzett az ügyfél. A jegy állapotát “folyamatban lévő”-ről “elkészült”-re állítja át. Ekkor a jegy visszakerül az ügyfélhez, aki validálni fogja, hogy tényleg megtörtént-e a probléma javítása.

### Kivételes eset: Ha az ügyél úgy gondolja, nem történt meg a javítás, a jegyet újra kell nyitni.

## Használati eset 15

### *Név:* Folyamatban lévő hibajegy/kérés várakozó állapotra állítása

### Aktor: Fejlesztő

### Cél: A fejlesztő jóvá szeretné hagyatni a jegy javítását.

### Bemenet: Egy folyamatban lévő hibajegy.

### Kimenet: A bemeneti hibajegy állapota várakozó.

### Leírás: A javítás megkezdése előtt a hibajegyet jóvá kell hagyatni egy jóváhagyó szerepkörű felhasználóval.

### Kivételes eset: -

## Használati eset 16

### *Név:* Várakozó hibajegy/kérés visszaállítása folyamatban lévő állapotra

### Aktor: Jóváhagyó

### Cél: A hiba javítását engedélyezni a fejlesztő számára.

### Bemenet: Egy várakozó állapotban lévő hibajegy.

### Kimenet: A bemeneti hibajegy állapota “folyamatban lévő”.

### Leírás: A jóváhagyú szerepkörű felhasználó jelzi, hogy a hibajegy javítása ténylegesen a fejlesztők feladata, vagy esetlegesen rajtuk kívül álló a hiba oka, így nem foglalkoznak a javításával.

### Kivételes eset: Ha a jóváhagyó mégsem engedélyezi a hibajegy javítását, annak állapota “elutasítva” lesz.

## Használati eset 17

### *Név:* Hibajegyhez megjegyzés fűzése

### Aktor: Fejlesztő / Jóváhagyó / Ügyfél

### Cél: Egy kommunikációs felület biztosítása a különböző szerepkörű felhasználók között.

### Bemenet: Egy hibajegy.

### Kimenet: A hibajegyhez egy új megjegyzés került hozzáfűzésre.

### Leírás: A különböző szerepkörű felhasználók megegyzések írásával jelezhetik döntéseik okát. Pl.: a jóváhagyó egy megjegyzésben írhatja le, miért utasította el a hibajegy javításának megkezdését.

**Kivételes eset:** -

# Nem funkcionális követelmények

Ebbe a kategóriába tartozik annak megkötése, hogy csak bejelentkezett felhasználók használhassák a rendszert. Az egyes bejelentkezett user-ek csak azon projektekben tevékenykedhetnek, melyekhez egy adminisztrátor hozzárendelte őket. Fontos, hogy a skálázhatóság tekintve képes legyen a rendszer egyszerre akár 10 felhasználót is gond nélkül kiszolgálni. A felhasználói felület nyelvezete legyen magyar.