

Tudu-ToDo - tesztelési dokumentáció

1. Felhasználói tesztelés:

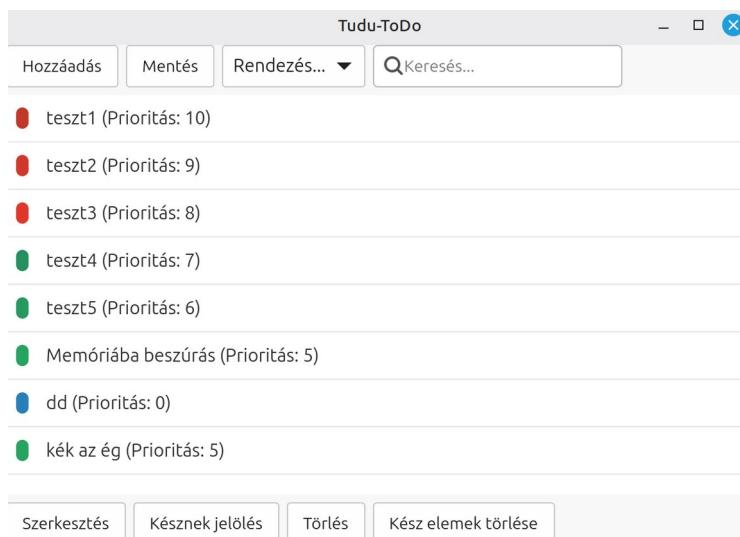
Az alábbi felhasználói (manuális) tesztelési dokumentáció a specifikáció és a projekt tesztjei alapján. minden teszthez rövid leírás, lépések, várt eredmény és pass/fail kritérium.

- Tesztelő neve: Fehér Vivien Fanni
- Ajánlott környezet:
 - OS: Linux (Debian/Ubuntu ajánlott), vagy bármilyen GTK4-támogatott rendszer.
 - Processzor: Bármilyen (x86_64, ARM ajánlott).
 - RAM: minimum 512 MB (Xvfb + GTK + alkalmazás).
 - Tárhely: minimum 1 GB (fordítás, Criterion, adatfájlok).

Előfeltételek:

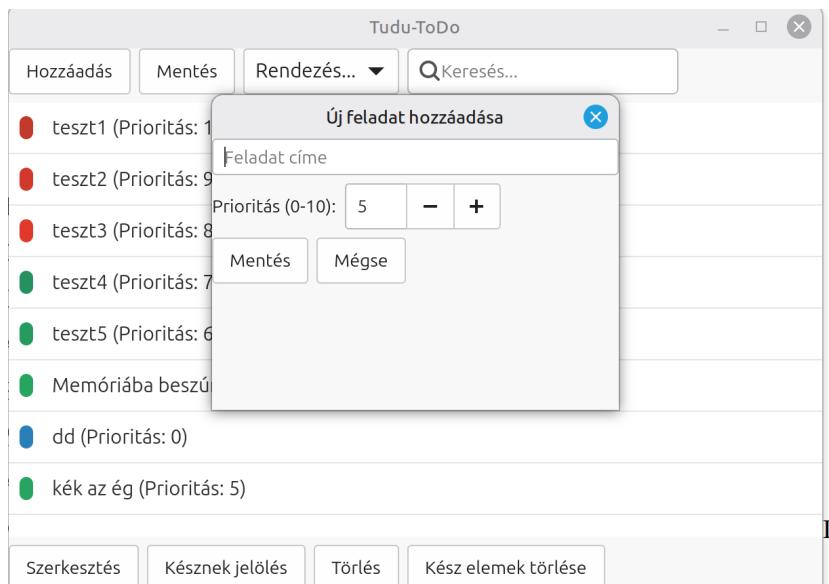
1) Alapindítás / főablak:

- Cél: a főablak megnyílik, a felület elemei láthatók.
- Lépések:
 1. Alkalmazás elindítása
 2. Ellenőrzés: ablak megjelenik, cím, hozzáadás/gombok/lista/search/sort mezők láthatók.
 - Várt eredmény: ablak nyílik, fő vezérlők elérhetők.
 - Pass: minden felület elem megjelenik és nem dob azonnali hibát.



2) Elem hozzáadása (Hozzáadás):

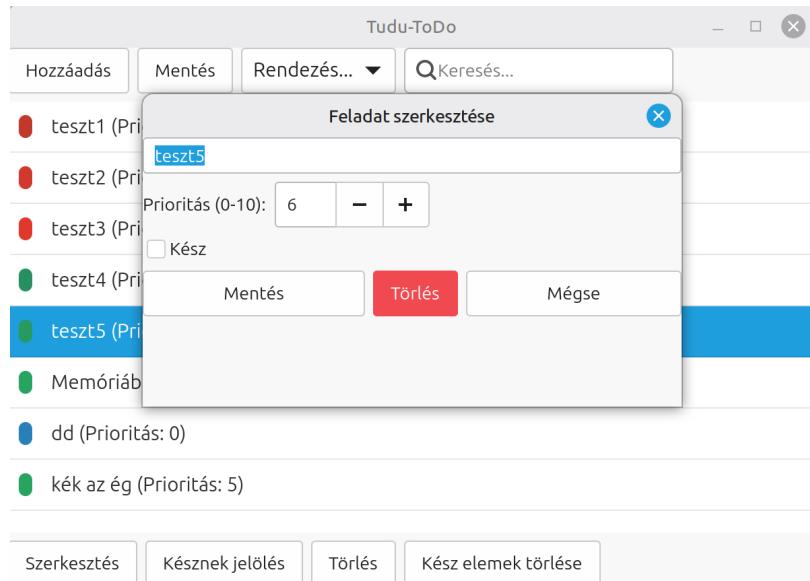
- Cél: új todo létrehozása és lista frissítése.
- Lépések:
 1. A "Hozzáadás" gomb megnyomása.
 2. Kitölteni a címet, leírást, prioritást, kategóriát.
 3. Mentés/OK.
 4. A lista megnézése: megjelent-e új sor.
 - Várt eredmény: új téTEL jelenik meg a listában; az adatok elérhetők (a GUI-ban: cím, prioritás).
 - Pass: téTEL megjelenik, helyes adatokkal.



3) Elem szerkesztése (Szerkesztés):

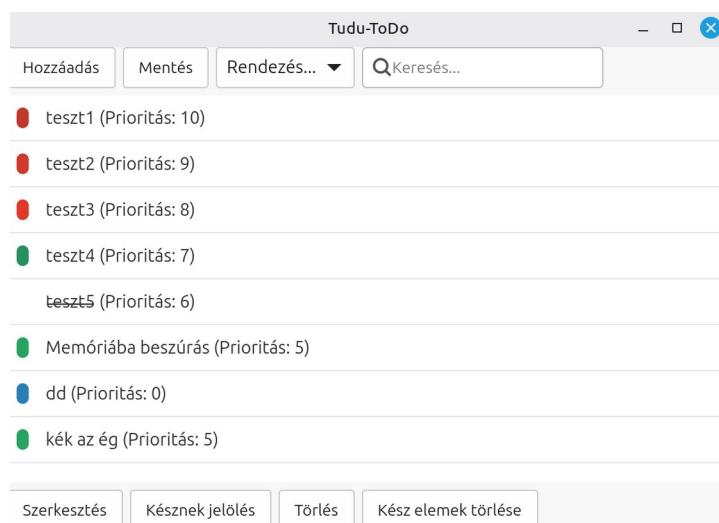
- Cél: meglévő téTEL módosítható és az ablak viselkedése.
- Lépések:
 1. Kiválasztani egy elemet a listából.
 2. Megnyomni a "Szerkesztés" gombot.
 3. Módosítani a szöveget/priority-t, aztán mentés.
 4. Ismét a "Szerkesztés"-re kattintani ugyanarra az elemre — ellenőrizni, hogy csak egy szerkesztés ablak nyílik-e (singleton).

- Várt eredmény: módosítások látszódnak a listában; egyfél szerkesztés-ablak referencia létezik.
- Pass: adatok frissülnek, szerkesztés ablak singleton.



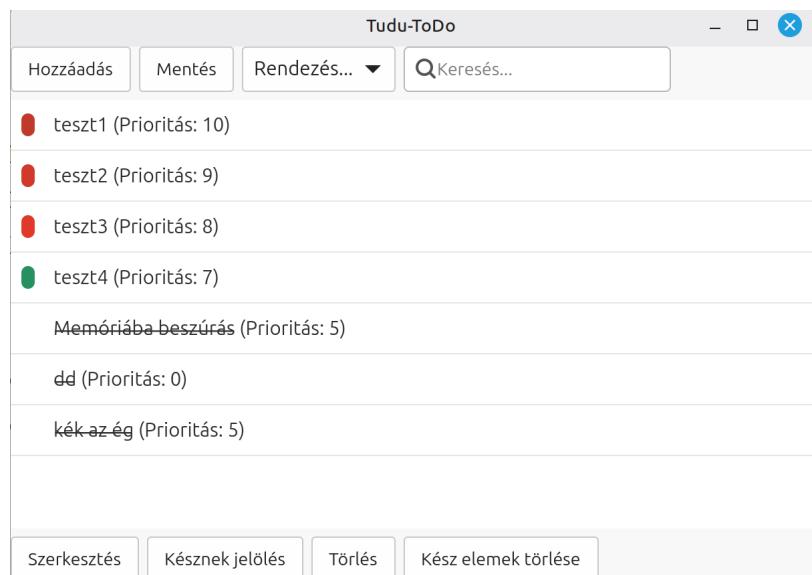
4) Készre jelölés (Késznek jelölés):

- Cél: térel státuszának váltása.
- Lépések:
 1. Kiválasztani egy elemet.
 2. Megnyomni a "Késznek jelölés".
 3. Ellenőrizni, hogy a térel jelölve (áthúzott).
 - Várt eredmény: vizuális és adatbeli státuszváltás.
 - Pass: státusz megváltozik, mentéskor is megmarad.



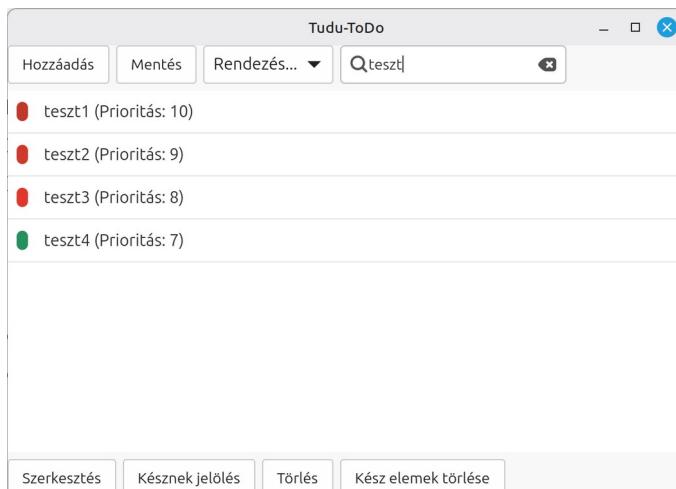
5) Törlés és kész elemek törlése:

- Cél: térel(ek) törlése, törlés csak kész tételek közül.
- Lépések:
 1. Választani egy tért, a „Törlés” gomb megnyomása.
 2. Jelölni a kész tételeket, majd „Kész elemek törlése”.
 - Várt eredmény: törölt elemek eltűnnek, fájlba mentés után nem jelennek meg.
 - Pass: törlés végrehajtódik, nincs maradék referencia.



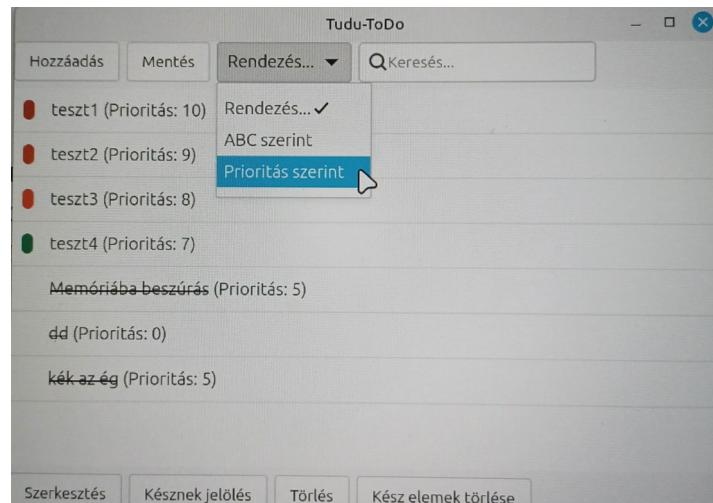
6) Keresés (Search):

- Cél: keresés működése
- Lépések:
 1. Megadni egy keresőszót a „Keresés” mezőbe.
 2. Figyelni a lista szűrését.
 3. Ellenőrizd több találat, nulla találat esetét.
 - Várt eredmény: csak a találatok látszanak; nulla találatnál üres lista.
 - Pass: keresés megfelelő találatokat ad.



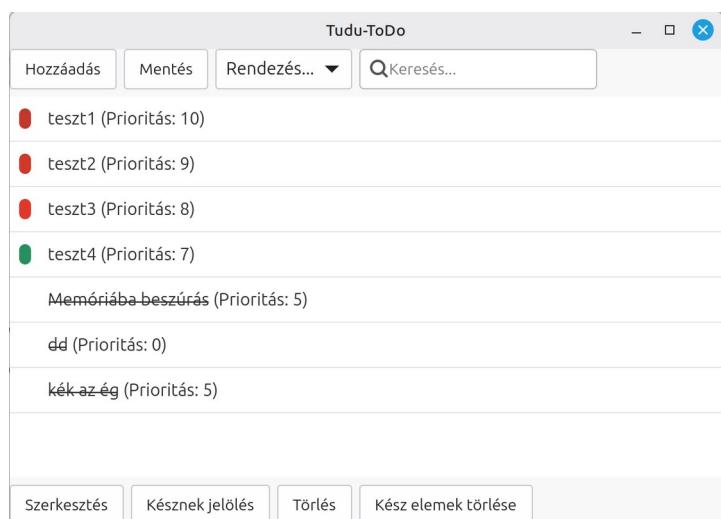
7) Rendezés (Sort):

- Cél: prioritás/ABC sorrend szerinti rendezés működése.
- Lépések:
 1. Választani egy rendezés mezőt (prioritás, ABC).
 2. Ellenőrizni a lista sorrendjét.
 - Várt eredmény: lista rendezett a kiválasztott kritérium szerint.
 - Pass: rendezés helyes.



8) Mentés / Betöltés:

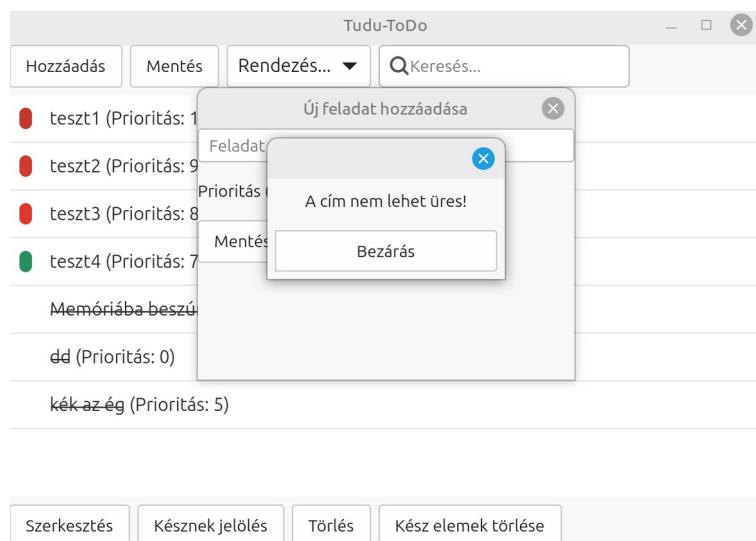
- Cél: lista fájlba mentése és visszatöltése.
- Lépések:
 1. Több elem létrehozása.
 2. Mentés gomb.
 3. A todos.csv fájl megnyitása.
 4. Ellenőrizni, hogy az adatok sikeresen beírásra kerültek a csv fájlba.
 - Várt eredmény: az összes mező és elem mentésre került.
 - Pass: mentés -> betöltés után azonos elemek.



todos.csv > data	
1	teszt1 10 0
2	teszt2 9 0
3	teszt3 8 0
4	teszt4 7 0
5	Memoriába beszúrás 5 1
6	dd 0 1
7	kék az ég 5 1
8	

9) Üzenetek / hibaablakok:

- Cél: felhasználói hibák/figyelmeztetések helyes kezelése (üres cím, fájl hiba).
- Lépések:
 1. Üres címmel menteni egy elemet.
 - Várt eredmény: érthető hibaüzenet jelenik meg; alkalmazás nem omlik össze.
 - Pass: hibaüzenet és stabilitás.



2. Fejlesztői tesztelés:

1) Rövid áttekintés:

A tesztek Criterion keretrendszerrel készültek. A források a tests/ könyvtárban vannak; a projekt valós implementációit a src/ tartalmazza (model, view, search, controller,extras, io). A view- és GTK-tesztek grafikus környezetet igényelnek (Xvfb-vel futtatva).

2) Függőségek:

- build tools: gcc, make
 - Criterion (libcriterion-dev)
 - GLib/GIO/GObject, GTK4 (development csomagok)
 - Xvfb (headless futtatóhoz)
-
- Telepítés Linux Mint alatt:
 - sudo apt update

- sudo apt install -y build-essential libglib2.0-dev libgtk-4-dev libgirepository1.0-dev libcriterion-dev xvfb

3) Futtatás:

1. Fordítás:

- make -C tests clean
- make -C tests
- ./tests/run_tests

2. GTK-tesztek:

- xvfb-run --auto-servernum --server-args=''-screen 0 1024x768x24' ./tests/run_tests

4) Tesztfájlok és céljuk:

- test_controller.c:
 - controller logika, init/shutdown hívások ellenőrzése.
- test_extras.c:
 - segédfüggvények (extras) viselkedése.
- test_hello.c (próba fájl a tesztelés elindításához):
 - egyszerű sanity-check.
- test_io.c:
 - fájl be/ki olvasás, IO viselkedés.
- test_model.c:
 - model_create_list, model_add_item, model_remove_item, model_free_list.
- test_search.c:
 - search_filter_wrapper: szűrési logika.
- test_view.c:
 - view_create_main_window, view_refresh_list, view_create_add_window/edit_window singleton-viselkedés.

5) A teszt eredményének kimenete:

- Siker: minden teszt zöld, pl. "[====] Synthesis: Tested: N | Passing: N | Failing: 0 | Crashing: 0".

- **A teszt logjának eredménye: minden teszt sikeres volt.**

```
$ ./tests/run_tests
make: Nothing to be done for 'all'.
Running test_controller
[====] Synthesis: Tested: 2 | Passing: 2 | Failing: 0 | Crashing: 0
Running test_extras
[====] Synthesis: Tested: 6 | Passing: 6 | Failing: 0 | Crashing: 0
Running test_hello
[====] Synthesis: Tested: 1 | Passing: 1 | Failing: 0 | Crashing: 0
Running test_io
A(z) /tmp/tudu_io_test_nonexistent.tsv fajl nem talalhato, ures lista betolt
ve.
Lista elmentve a(z) /tmp/tudu_io_test.tsv fajlba.
2 elem betoltve a(z) /tmp/tudu_io_test.tsv fajlbol.
[====] Synthesis: Tested: 2 | Passing: 2 | Failing: 0 | Crashing: 0
Running test_model
[====] Synthesis: Tested: 4 | Passing: 4 | Failing: 0 | Crashing: 0
Running test_search
[====] Synthesis: Tested: 3 | Passing: 3 | Failing: 0 | Crashing: 0
Running test_view
[====] Synthesis: Tested: 2 | Passing: 2 | Failing: 0 | Crashing: 0
```

- Általános figyelmeztések:
 - "Unable to acquire session bus: A megadott cím üres" — akkor jelenik meg ha DBUS_SESSION_BUS_ADDRESS üresre van állítva a teszben; ártalmatlan.
 - GTK figyelmeztések headless backend hiánya — Xvfb-t használni.

6) Gyakori hibák és javítások:

- Segfault a view/main teszteknél:
 - Ok: GtkApplication/GApplication nincs regisztrálva.
 - Javítás: regisztrálni az app-ot a teszben:
g_application_register(G_APPLICATION(app), NULL, NULL); illetve adni érvényes GtkApplication példányt.
- "No such backend: headless":
 - Ok: GDK headless backend nincs telepítve.
 - Javítás: futtatni az Xvfb alatt (xvfb-run ...) és telepíteni a megfelelő GDK backendet.
- Linker hibák (multiple definition):
 - Ok: stubs és src egyszerre definiálják ugyanazt a függvényt.
 - Javítás: a test_stubs.c fájlban csak deklarációkat hagyni, ne implementációkat azokból a függvényekből, amelyek implementálva vannak src/ alatt.
- Timeout vagy "Failed to register":

- Ok: GApplication megpróbál D-Bus-on regisztrálni.
- Javítás: tesztkörben kerülni a DBUS autolaunch-t.

7) Hibakeresés (debug lépések):

- Ha olyan hiba lép fel amihez szükséges részletesebb log, akkor ennek a parancsnak a segítségével lehet részletesebb hibaleírást kapni. Erre szükség volt teszteléskor:
 - gdb --args ./tests/test_view
 - run
 - bt full

8) Új teszt hozzáadása — rövid útmutató:

1. tests/ könyvtárban új fájl test_myfeature.c létrehozása, Criterion tesztesetekkel.
2. Beleírni a tests/Makefile SOURCES sorába a fájl nevét.
3. Build + futtatás:
 - make -C tests clean
 - make -C tests
 - ./tests/run_tests