



**SZÉCHENYI
EGYETEM**
UNIVERSITY OF GYŐR



**INFORMATIKA
TANSZÉK**
DEPARTMENT OF INFORMATICS

Ipar 4.0 technológiák beadandó

Smart Strength Machine SysML alapú
modellezése

Vass József

Gazdaságinformatikus BSc

2025/2026/1

Tartalom

1. Bevezetés.....	1
2. Rendszer célja és megoldandó probléma	1
3. Érintettek (stakeholderek) azonosítása és jellemzése	2
4. Követelmények	3
5. Használati esetek.....	8
6. Struktúra modellek és parametrikus diagram.....	12
7. Rendszer viselkedése	19
8. Összefoglalás	26
Ábrajegyzék	27
Irodalomjegyzék.....	27

1. Bevezetés

A sporttechnológia fejlődésének egyik iránya az „okos edzőgépek” megjelenése, amelyek képesek mérni a sportoló mozgását, erőkifejtését, és a begyűjtött adatok alapján valós időben módosítani az edzés paramétereit. A beadandóban egy ilyen rendszer modellezésével foglalkozom. A választott példa egy **Smart Bench Press**, vagyis egy olyan fekvenyomó gép, amely digitális vezérléssel, szenzorokkal és ellenállásszabályzó egységgel rendelkezik.

A különböző sportokban különböző edzéshatásokra van szükség (pl maximális erő, hipertrófia, gyors erő fejlesztése). Ezenkívül a mozdulat különböző fázisaiban erőkartól függően változik a sportoló által kifejtendő erő mértéke illetve ismétlésről ismétlésre egyre kisebb erőt képes kifejteni. Ezért az optimális terheléshez szükség van az alkalmazkodó ellenállásra, hogy folyamatosan biztosítva legyen a kívánt edzéshatáshoz szükséges ellenállás. A gép célja, hogy támogassa a sportolót a számára kívánt edzéshatás hatékonyabb elérésben.

A gép működésének alapja, hogy a felhasználó kiválaszt egy kezdő ellenállást és egy **nehézségi szintet 1 és 10 között**, amelyhez a rendszer automatikusan hozzárendel egy úgynevezett célsebességet. Edzés közben a gép folyamatosan méri a rúd tényleges sebességét, és ha a sportoló gyorsabban mozog a kívánt tempónál, akkor a rendszer automatikusan növeli az ellenállást. Ha pedig lassabban mozog, akkor csökkenti azt. Így a gép mindig olyan ellenállást tart fenn, amely a megadott nehézségi szinthez tartozó sebességhez illeszkedik. A rendszer nemcsak valós idejű visszajelzést ad a kijelzőn keresztül, hanem csatlakoztatható egy mobilalkalmazás is, amelybe a sportoló a saját felhasználói fiókjába elküldheti és elmentheti az edzés adatait későbbi elemzés céljából.

A beadandó célja a Smart Bench Press rendszer **modellezése SysML nyelv segítségével**. Ennek során a dokumentumban:

- meghatározom a rendszer célját és a megoldandó problémákat,
- azonosítom az érintetteket,
- összegyűjtöm és modellezem a rendszer követelményeket,
- összegyűjtöm és modellezem a használati eseteit,
- megalkotom struktúra modelleket (BDD, IBD), és létrehozom a parametrikus diagramot (PAR),
- definiálom a rendszer viselkedést (Activity, Sequence, State),

A beadandó célja, hogy egy valószerű, modern sporttechnológiai példán keresztül bemutassa, hogyan használható a SysML egy összetettebb, fizikai és digitális komponensekből álló rendszer áttekinthető és következetes leírására.

2. Rendszer célja és megoldandó probléma

A rendszer célja:

A Smart Bench Press rendszer célja, hogy egy hagyományos fekvőnyomó gépet digitálisan támogatott, kiber-fizikai eszközzé alakítsa, amely képes valós időben igazítani az ellenállást a sportoló mozgási sebessége alapján. A rendszer célja olyan edzés biztosítása, amely egyszerre biztonságosabb, hatékonyabb és személyre szabottabb, mint a klasszikus súlyokkal történő edzés. További cél biztosítani az edzés adatok mobilalkalmazásban való tárolásának lehetőségét.

Megoldandó probléma:

1. **A hagyományos súlyok nem képesek valós időben (pillanatról pillanatra) alkalmazkodni a sportoló aktuális állapotához és a mozdulat különböző tartományaihoz**, így nem biztosítanak optimális terhelést minden ismétlés során.
2. **Az edzések valódi intenzitását nehéz objektíven mérni**, mert a sportoló nem látja a pontos sebességét, erő kifejtését.
3. **Az edzésadatok gyűjtése körülményes**, ezért sokszor elmarad, így nincs lehetőség az edzésterv hosszú távú követésére.

A rendszer által kínált megoldás:

- A rendszer a rúdmozgásának sebességét folyamatosan méri a gyakorlat közben, és összeveti egy előre meghatározott célsebességgel.
- Szükség esetén a gép növeli vagy csökkenti az ellenállást annak érdekében, hogy a sportoló a kívánt tempót tartsa.
- A rendszer valós időben mutatja az ellenállás aktuális értékét.
- Az edzésadatait automatikusan elmenti, és feltölti a felhasználó mobilalkalmazás fiókjába.

3. Érintettek (stakeholderek) azonosítása és jellemzése

A rendszer fejlesztése során fontos meghatározni azokat a személyeket, csoportokat és szervezeteket, akik valamilyen módon kapcsolatba kerülnek a Smart Bench Press működéssel, vagy érintettek annak eredményeiben. Az érintettek eltérő célokkal, elvárásokkal és érdekeltségekkel rendelkezhetnek, ezért az ő szempontjaik megértése alapvető a rendszer követelményeinek kialakításához. Az alábbiakban felsorolom a releváns stakeholder szerepköröket, és röviden összefoglalom, hogy mi az érdekük a rendszerrel kapcsolatban. [1]

Stakeholder szerepkörök és érdekeik:

Stakeholder	Érdeke a rendszerrel kapcsolatban
Sportoló	Biztonságos, személyre szabott terhelést szeretne valós idejű visszacsatolással. Az edzésadatait szeretné automatikusan feljegyezni, hogy követni tudja a fejlődését.

Fitneszterem üzemeltetője	Modern, vonzó, karbantartható eszköz biztosítása. Hibamentes működés, adatbiztonság.
Fejlesztők	Átlátható, egyértelmű követelmények és jól definiált rendszerhatárok.
Mobilalkalmazás	Hibamentes és biztonságos kommunikáció, adatátvitel.
Karbantartó személyzet	Diagnosztikai információk elérése.

4. Követelmények

A követelményeket három szinten fogalmazom meg:

- **Magas szintű követelmények (HLR)** amelyek leírják „mit” szeretnének az érintettek és még fekete-doboz jellegűek.
- **Derivált követelmények (DR)** amelyek segítenek a fekete-doboz jellegű HLR-ekből egyértelmű, pontos LLR-eket létrehozni.
- **Alacsonyszintű követelmények (LLR)** amelyek implementáció-közeli, „hogyan” jellegű követelmények. [2]

A követelmények táblázatos formában jelennek meg. Minden követelményhez ID-t, kapcsolatot, nevet, leírást, prioritást és validációt rendelek.

A Kapcsolat oszlop azt mutatja meg, hogy az adott követelmény melyik magasabb szintű követelményből származik. HLR esetében ilyen nincs.

A Prioritás oszlop azt jelzi, hogy az adott követelmény milyen mértékben kritikus a rendszer működése szempontjából:

- **P1 - Magas prioritás:** A rendszer alapvető működése, biztonsága vagy megbízhatósága szempontjából kritikus követelmény.
- **P2 - Közepes prioritás:** Fontos funkció, amely hozzájárul a rendszer megfelelő használhatóságához vagy adatkezeléséhez, de nem biztonságkritikus.
- **P3 - Alacsony prioritás:** Opcionális vagy kényelmi jellegű követelmény, amely javítja a felhasználói élményt vagy a diagnosztikát, de elhagyása nem veszélyezteti a rendszer működését.

A követelménytáblázat Validáció oszlopa azt írja le, hogy **hogyan lehetne igazolni**, hogy az adott követelmény teljesült-e.

Végül a követelményeket követelménydiagramon (1. ábra) is ábrázolom.

Magas szintű követelmények (HLR):

ID	Név	Leírás	Validáció
	Prioritás		
HLR-1	Személyre szabható edzésszint	A rendszernek lehetővé kell tennie a felhasználó számára a kezdő ellenállás és a gyakorlat célsebességének beállítását 1-10 skálán.	A beállítás után a rendszer a valós nehézségi szintet biztosítja.
	P1		
HLR-2	Automatikus ellenállás szabályozás	A rendszernek a mért sebesség alapján automatikusan állítania kell az ellenállást.	Lassabb és gyorsabb mozgást szimulálva, az ellenállás és ezáltal a sebesség növekszik vagy csökken a logikának megfelelően.
	P1		
HLR-3	Valós idejű visszajelzés	A rendszernek valós időben meg kell jelenítenie a pillanatnyi terhelést, sebességet és ismétlésszámot.	Edzés közben vizuálisan megjelenő értékek azonnal frissülnek.
	P2		
HLR-4	Edzésadatok mentése	A rendszernek el kell mentenie az edzés során keletkezett adatokat a mobilalkalmazásban a csatlakoztatott felhasználói fiókban.	Edzés lefuttatása után a mobilalkalmazás megkapja az adatokat.
	P2		
HLR-5	Biztonságos működés	A rendszernek meg kell akadályoznia a veszélyes vagy hirtelen terhelésváltozásokat.	Extrém sebességváltozásokat szimulálva a rendszer nem vált túl nagy terhelésre.
	P1		
HLR-6	Diagnosztikai támogatás	A rendszernek alapvető hibakódokat és diagnosztikai adatokat kell elérhetővé tennie	A hibapló megtekintése során tartalmazza a releváns diagnosztikai információkat.
	P3		

Derivált követelmények (DR):

ID	Név	Kapcsolat	Leírás	Validáció
		Prioritás		
DR-1	Nehézségi szintekhez tartozó célparaméterek	HLR-1	A rendszernek minden nehézségi szinthez (1–10) hozzá kell rendelnie egy célsebességet. 1 nagyon lassan, 10 nagyon gyorsan mozgatható a rúd.	Konfigurációs táblázatban ellenőrizhető, hogy mind a 10 szinthez tartozik célsebesség.
		P1		
DR-2	Hibahatár definiálása a beállított szinthez	HLR-1	A rendszernek definiálnia kell egy hibahatárt, amellyel a rúd kezdő ellenállása és sebessége eltérhet a beállítottól.	A specifikációban vagy konfigurációban ellenőrizhető, hogy a hibahatár meg van adva (pl. +/-10%).
		P1		
DR-3	Rúdpozíció mérése	HLR-2	A rendszernek a megfelelő időközönként	A rudat mozgatva ellenőrizni kell, hogy a
		P1		

			mérnie kell a rúd pozícióját és az időpontot.	rendszer valós időben frissíti-e a pozícióértékeket.
DR-4	Rúdpozíciók összehasonlítása	HLR-2	A rendszernek az egymás utáni rúdpozíció adatokat kell sebességméréshez használnia.	Ellenőrizni kell a sebességmérési logikát.
		P1		
DR-5	Maximális ellenállásváltozási ráta	HLR-2, HLR-5	Az ellenállásváltozás csak fokozatosan történhet.	Hirtelen sebességváltozást szimulálva ellenőrizni kell, hogy a terhelés nem nő vagy csökken ugrásszerűen.
		P1		
DR-6	Kommunikációs hiba észlelése	HLR-4	A rendszernek észlelnie és jeleznie kell, ha a mobilalkalmazással megszakad a kapcsolat.	Manuálisan megszakítjuk a kapcsolatot, és ellenőrizzük, hogy hibajelzés történik-e.
		P2		
DR-7	Hibás mérések naplózása	HLR-6	A rendszernek naplóznia kell a nem megfelelő működést.	A hibanapló ellenőrzése során meg kell jelennie a nem megfelelő működést jelző adatoknak.
		P3		
DR-8	Megjelenítési frissítési ciklus definiálása	HLR-3	A rendszernek definiálnia kell egy frissítési ciklusidőt, amely meghatározza, milyen gyakran frissülnek a kijelzőn megjelenő edzésparaméterek.	A specifikációban ellenőrizhető, hogy a frissítési ciklus dokumentálva van.
		P2		
DR-9	Megjelenítendő adatok adatszerkezetének meghatározása	HLR-3	A rendszernek definiálnia kell egy egységes adatszerkezetet a kijelzőn való megjelenéshez (pl. aktuális ellenállás, aktuális sebesség, ismétlésszám).	A dokumentációban meg kell jelennie a kijelző paramétereit tartalmazó adatszerkezetnek.
		P2		

Alacsony szintű követelmények (LLR):

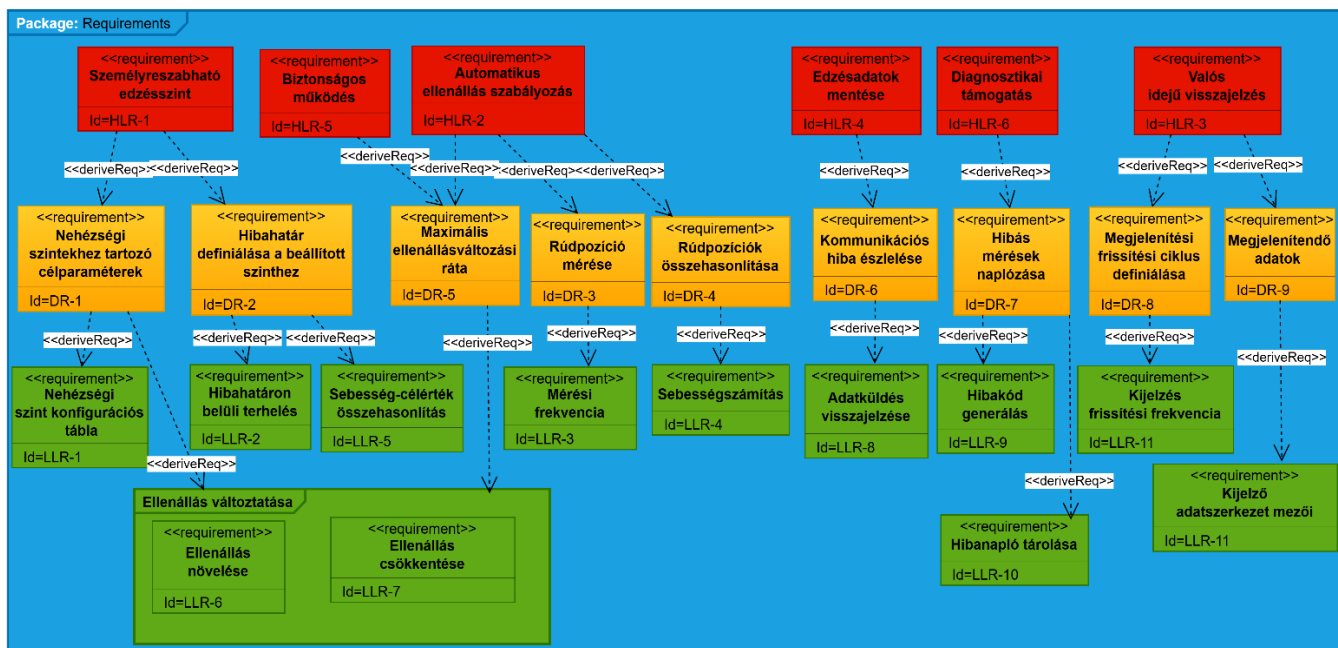
ID	Név	Kapcsolat	Leírás	Validáció
		Prioritás		
LLR-1	Nehézségi szint konfigurációs tábla.	DR-1	A rendszernek egy konfigurációs táblában kell tárolnia a 10 nehézségi	A konfigurációs fájl ellenőrzése
		P1		

			szinthez tartozó célsebességet	
LLR-2	Hibahatáron belüli terhelés	DR-2	A rendszernek biztosítania kell, hogy a tényleges sebesség eltérése a célértéktől legfeljebb +/-10% lehet.	Adatelemzés edzésadatokon.
		P1		
LLR-3	Mérési frekvencia	DR-3	A rendszernek 50 ms-os (+/-10%) időközönként frissítenie kell a pozícióadatokat.	Logadatok elemzése
		P1		
LLR-4	Sebességszámítás	DR-4	A sebességet a pozícióváltozás alapján kell meghatározni. $v=(x_2-x_1)/(t_2-t_1)$	Ellenőrizni kell a vezérlő modul számítási logikáját dokumentáció alapján.
		P2		
LLR-5	Sebesség-célérték összehasonlítás	DR-2	A rendszernek minden ciklusban össze kell vetnie a mért sebességet a célsebességgel. Ahol a hiba nem haladhatja meg a +/-2%-ot	Lassú/gyors mozgást szimulálva ellenőrizhető, hogy a rendszer helyesen érzékeli az eltérést.
		P1		
LLR-6	Ellenállás növelése	DR-1, DR-5	A célsebesség alatti mozgás esetén a rendszernek növelnie kell az ellenállást, a korábbi ellenállás maximum 10%-val.	Szándékosan lassabb mozgás esetén ellenőrizni kell az ellenállás növekedését.
		P1		
LLR-7	Ellenállás csökkentése	DR-1, DR-5	A célsebesség feletti mozgás esetén a rendszernek csökkentenie kell az ellenállást a korábbi ellenállás maximum 10%-ával.	Gyorsabb mozgás esetén ellenőrizni kell az ellenállás csökkenését.
		P1		
LLR-8	Adatküldés visszajelzése	DR-6	A rendszernek az edzésadatok elküldése után 1 másodpercen belül állapotüzenetet kell küldenie (OK / ERROR) a mobilalkalmazás felé.	Adatátvitel után a mobilapp visszajelzését kell ellenőrizni.
		P2		
LLR-9	Hibakód generálás	DR-7	A rendszernek hibakódot kell generálnia 100 ms-on belül, a következő kódkategóriák egyikébe sorolva:	Hiba injektálása.
		P2		

			kommunikációs hiba, szenzorhiba, logikai hiba.	
LLR-10	Hibanapló tárolása	DR-7	A rendszernek tárolnia kell az utolsó 10 hibát diagnosztikai célból. Mindegyiknek tartalmaznia kell időbélyeget, hibakódot, forrásmodult.	Naplófájl ellenőrzése.
		P3		
LLR-11	Kijelzés frissítési frekvencia	DR-8	A rendszernek biztosítani kell, hogy a kijelzőn megjelenő értékek legfeljebb 100 ms-onként frissüljenek.	Frissítési idő mérése.
		P2		
LLR-12	Kijelző adatszerkezet mezői	DR-9	A kijelzéshez használt adatszerkezetnek legalább három mezőt kell tartalmaznia: aktuális ellenállás, sebesség, ismétlésszám.	Ellenőrizhető, hogy a kötelező mezők megjelennek.
		P2		

A 10 nehézségi szintet tartalmazó konfigurációs tábla:

Szint	Edzéstípus fókusz	Célsebesség (m/s)
1	Maximálisan robbanékony	1.2
2	Robbanékony erő magas sebességgel	1
3	Dinamikus / robbanékony erő	0.85
4	Vegyes: hipertrófia, dinamika	0.7
5	Hipertrófia / erő-állóképesség	0.55
6	Hipertrófia - közepes	0.45
7	Hipertrófia (izomtömeg) - lassabb	0.35
8	Erő-hipertrófia	0.25
9	Max erő / erő-hipertrófia	0.2
10	Max erő, nagyon lassú	0.15



1. Requirement Diagram (saját szerkesztés)

5. Használati esetek

A használati esetek összegyűjtését a következő feladatok teljesítésével fogom végrehajtani:

1. **A rendszerhatár kijelölése**, ahol a rendszer kapcsolódik a külvilággal.
2. **Aktorok azonosítása**, akik közvetlen kapcsolatba lépnek a rendszerrel.
3. **Use case-ek azonosítása**, amelyek mindegyike egy aktor cél, hogy mit akar elérni az aktor a rendszerrel aktuálisan.
4. **Use case forgatókönyvek**, amelyek a használati esetek szöveges leírásai és tartalmazni fogják a prekondíciókat, posztkondíció(ka)t, főfolyamot és alternatív folyamo(ka)t.

5.1 A rendszerhatár kijelölése

A rendszerhatár azt jelöli ki, hogy a rendszer milyen képességeket tartalmaz amelyeken keresztül kapcsolatba lép a külvilággal. A HLR-ek alapján a rendszer ilyen funkciói a következők:

- Kezdő ellenállás és a sebesség megadásának biztosítása.
- Ellenállás és sebesség mérése, és az ellenállás biztonságos mértékű állítása.
- Adatok (ellenállás, sebesség, ismétlésszám) megjelenítése.
- Kapcsolat létesítése mobilalkalmazással, és adatokat küldeni a számára.
- Hibakódok és diagnosztikai adatok gyűjtése és megjelenítése.

Az aktorok azok a rendszerhatáron kívüli szereplők lesznek akik közvetlen kapcsolatba kerülnek a fenti képességekkel.

5.2 Aktorok azonosítása

A fenti képességek alapján 3 fő aktor vesz részt a rendszerben:

Aktor	Leírás	Mit csinál a rendszerrel?
Sportoló (Primary Actor)	A gépet használja edzésre.	Beállítja a nehézségi szintet, edzést végez, visszajelzést figyel.
Mobilalkalmazás (External System)	A felhasználói fiók része.	Fogadja az edzésadatokat, visszaigazolást küld.
Karbantartó	Hibakeresésért felel.	Diagnosztikát, hibanaplót ellenőriz.

5.3 Use case-ek azonosítása

Minden use case egy aktor cél, hogy mit akar elérni az aktor a rendszerrel aktuálisan. A HLR-ek alapján a teljes UC-készlet:

Sportoló

- **UC1: Edzés végrehajtása**

Kapcsolódó HLR: HLR-1, HLR-2, HLR3

Leírás: A sportoló beállítja a kezdeti ellenállást és a célsebességet, majd végrehajtja az edzést. A folyamat során sportoló figyel az edzésadatokat (ellenállás, sebesség, ismétlésszám) és abbahagyja a gyakorlatot a kívánt edzésterhelés elérésekor.

Mobilalkalmazás

- **UC2: Edzésadatok mentése**

Kapcsolódó HLR: HLR-4

Leírás: A mobilalkalmazás a felhasználó fiókjával csatlakozik a rendszerhez. Az alkalmazás kezdeményezi az edzésadatok lekérését és fogadását, és elmenti az edzésadatokat a felhasználói fiókhoz tartozó adatbázisba. Ezután visszajelzést küld az mentés sikerességéről. A kapcsolat megszakadása esetén visszajelzést küld.

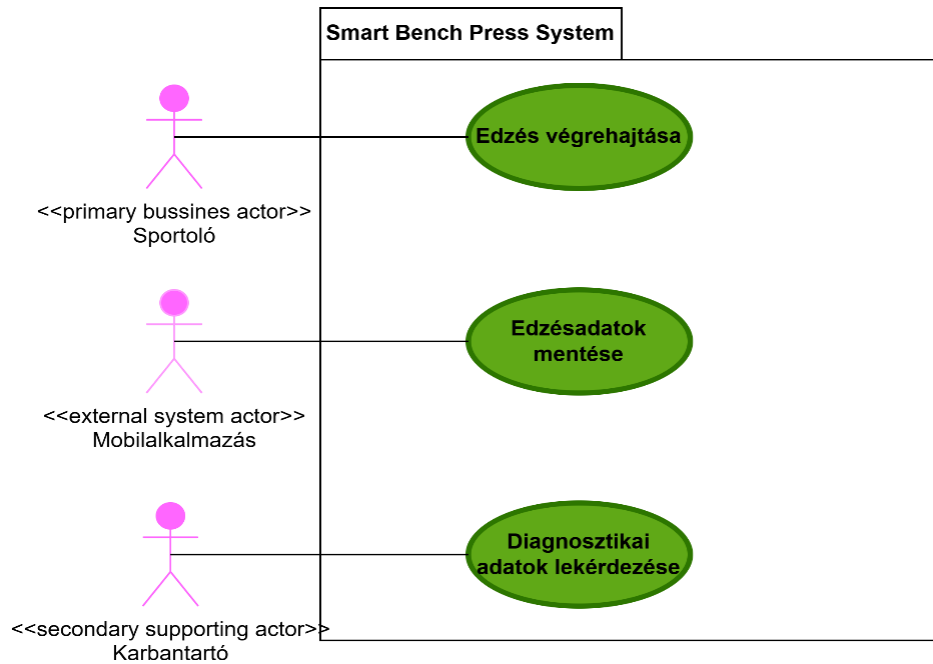
Karbantartó

- **UC3: Diagnosztikai adatok lekérdezése**

Kapcsolódó HLR: HLR-6

Leírás: A karbantartó megtekinti a hibanaplót és hibakódokat, amelyek segítik a hibakeresést és javítást.

A use case-ek és az aktorok kapcsolatát Use Case diagram (2. ábra) is szemlélteti. [3]



2. Use Case Diagram (saját szerkesztés)

5.4 Use case forgatókönyvek

5.4.1 UC1: Edzés végrehajtása

Cél: A sportoló végrehajt egy edzéssorozatot a kívánt terheléssel, valós idejű visszajelzéssel.

Prekondíciók:

- A Smart Bench Press be van kapcsolva, használatra kész.
- Nincs aktív hiba a rendszerben (nincs blokkoló hibakód).

Posztkondíció:

- Az edzéssorozat befejeződött.

Főfolyam (normál végrehajtás)

1. A **sportoló** a gép kezelőfelületén kiválasztja az „Edzés indítása” opciót.
2. A **rendszer** megjeleníti a nehézségi szintet (1–10) és a kezdő beállításokat.
3. A **sportoló** beállítja a kívánt nehézségi szintet (ellenállás + célsebesség).
4. A **rendszer** betölti a konfigurációs táblából a beállított szinthez tartozó célparamétereket.
5. A **sportoló** megkezdi a gyakorlatot (megfogja a rudat és elkezdi a mozgást).
6. A **rendszer** ciklikusan méri a rúd pozícióját és sebességét, és megjeleníti a pillanatnyi értékeket a kijelzőn (sebesség, ellenállás, ismétlésszám).
7. A **rendszer** a mért sebesség alapján automatikusan növeli vagy csökkenti az ellenállást, hogy a célsebességhez közelítsen.
8. A **sportoló** a kijelzőn figyeli az aktuális adatokat (ellenállás, sebesség, ismétlésszám) és folytatja a gyakorlatot.
9. Amikor a **sportoló** elérte a kívánt edzésterhelést, abbahagyja a gyakorlatot.

10. A **rendszer** lezárja az edzésszakaszt, és előkészíti az összesített edzésadatokat (összes ismétlés, átlagos sebesség, ellenállásprofil stb.).

Alternatív folyamatok

1 Érvénytelen nehézségi szint beállítása

- 3.1 A **sportoló** olyan értéket próbál megadni, amely kívül esik az ellenállás tartományon.
- 3.2 A **rendszer** hibaüzenetet jelenít meg: Érvénytelen nehézség.
- 3.3 A **rendszer** visszatér a nehézségi szint megadásához (2. lépés).

2 Edzés megszakítása a megkezdés előtt

- 5.1 A **sportoló** úgy dönt, hogy mégsem kezdi meg a gyakorlatot a megadott beállításokkal (pl. STOP gombot nyom).
- 5.2 A **rendszer** törli a beállításokat, és visszatér az edzés indítása pontra (1. lépés).

3 Ellenállási küszöb elérése

- 7.1 A **rendszer** nem képes tovább növelni, vagy csökkenteni az ellenállást, és jelzi a kijelzőn.
- 7.2 A **sportoló** dönthet úgy, hogy abbahagyja (9. lépés), vagy így is folytatja tovább a gyakorlatot (8. lépés).

5.4.2 UC2: Edzésadatok mentése

Cél: Az edzésadatok mentése a mobilalkalmazás adatbázisában a felhasználó fiókjához.

Prekondíciók:

- A sportoló edzésszakasza lezárult (UC1 befejeződött).
- Van elérhető hálózati kapcsolat.

Posztkondíciók:

- Sikeres esetben: az edzésadatai elérhetők a felhasználói fiókban a mobilalkalmazáson keresztül.
- Sikertelen esetben: a rendszer jelzi a hibát, és az edzésadatok lokálisan még elérhetők újraküldéshez.

Főfolyam (normál végrehajtás)

1. A **Smart Bench Press** előkészíti az edzésadatokat.
2. A **rendszer** ellenőrzi, hogy van-e kapcsolat mobilalkalmazással.
3. A **mobilalkalmazás** kapcsolatban áll a géppel (pl. Bluetooth vagy Wi-Fi).
4. A **rendszer** kezdeményezi az edzésadatok elküldését a mobilalkalmazás felé.
5. A **mobilalkalmazás** fogadja az adatokat és ellenőrzi, hogy azok formailag és tartalmilag érvényesek.
6. A **mobilalkalmazás** elmenti az adatokat a felhasználó fiókjához tartozó adatbázisba.
7. A **mobilalkalmazás** visszajelzést küld a Smart Bench Press felé az adatmentés sikerességéről.
8. A **rendszer** felajánlja új edzés indítását.

Alternatív folyamatok

1 Nincs kapcsolat a mobilalkalmazással

- 2.1 A **rendszer** figyelmeztet a mobilalkalmazás csatlakoztatására.
- 2.2 A **mobilalkalmazás** csatlakozik (3. lépés), vagy a **rendszer** új edzés indítását ajánlja fel (8. lépés).

2 Sikertelen mentés

- 6.1. A **mobilalkalmazás** hibaüzenetet küld a mentés sikertelenségéről.
- 6.2. Újrapróbálkozás (4. lépés) vagy a **rendszer** új edzés indítását ajánlja fel (8. lépés) és regisztrálja a hibát.

5.4.3 UC3: Diagnosztikai adatok lekérdezése

Cél: A karbantartó áttekinti a rendszerben tárolt hibakódokat és diagnosztikai adatokat.

Prekondíciók:

- A rendszer karbantartói módba kapcsolható.
- A hibanapló mentése aktív (LLR-10 teljesül).

Posztkondíciók:

- A karbantartó információhoz jut a hibák típusáról, előfordulási idejéről és forrásáról.
- A diagnosztikai információ felhasználható további hibakereséshez.

Főfolyam (normál végrehajtás)

1. A **karbantartó** belép a Smart Bench Press karbantartói felületére.
2. A **karbantartó** kiválasztja a „Hibanapló” megtekintése menüpontot.
3. A **rendszer** betölti az utolsó 10 hibanapló bejegyzést, a következő adatokkal: időbélyeg, hibakód kategória, forrásmodul.
4. A **rendszer** megjeleníti ezeket a karbantartó számára.
5. A **karbantartó** áttekinti a hibákat, és szükség esetén további lépéseket tesz.

Alternatív folyamatok

1 Nincs elérhető hiba a naplóban

- 3.1. A hibanapló üres (nem történt hiba).
- 3.2. A **rendszer** üzenetet jelenít meg: „Nem található naplózott hiba.”
- 3.3. A **karbantartó** kilép a rendszerből.

2 Hibanapló sérült vagy nem elérhető

- 3.1. A **rendszer** nem tudja beolvasni a hibanaplót.
- 3.2. A **rendszer** saját maga is hibát regisztrál (pl. „diagnosztikai alrendszer hiba”), és ezt megjeleníti a karbantartónak.

6. Struktúra modellek és parametrikus diagram

A struktúramodellek célja, hogy áttekinthetően és egyértelműen rögzítsék, hogyan épül fel a Smart Bench Press, és milyen belső szerkezet biztosítja a korábban meghatározott követelmények megvalósítását. [4]

A fejezet elkészítéséhez az alábbi feladatokat végzem el:

1. A rendszer fő komponenseinek azonosítása és hierarchiába rendezése (**Block Definition Diagram**).
2. A komponensek közötti kapcsolatok és adatáramlások feltérképezése (**Internal Block Diagram**).
3. A rendszer működése szempontjából fontos mennyiségek és képletek formalizálása, valamint ezek összekapcsolása a megfelelő komponensekkel (**Parametrikus diagram**).

6.1 Block Definition Diagram (BDD)

A Smart Bench Press rendszer felépítése több, egymástól elkülöníthető modulból áll, amelyek együttműködése biztosítja az automatikus ellenállásszabályozást, az edzésadatok valós idejű megjelenítését, valamint a mobilalkalmazással való kommunikációt. Az egyes blokkok a következő alrendszerekbe szerveződnek:

- **Smart Bench Press System** (fő blokk)
A teljes rendszer legfelső szintű blokkját képviseli. Magában foglal minden hardveres és szoftveres komponenst, valamint a kommunikációs és diagnosztikai alrendszereket. Ennek a blokknak a felelőssége, hogy a rendszer egészének működése összhangban legyen a HLR-ekben rögzített célokkal (pl. ellenállásszabályozás, valós idejű kijelzés, adatrögzítés). A moduláris felépítés lehetővé teszi, hogy a rendszer egyes részei külön fejleszthetők és tesztelhetők legyenek.
- **Szenzor modul**
A rúdpozíció és a rúd sebességének méréséért felel. Időbélyegzett pozícióadatokat szolgáltat a vezérlőegység számára. Az adatok pontos és stabil mérése alapfeltétele az automatikus ellenállásszabályozás működésének, mivel a DR-1, DR-3 és DR-4 követelmények mind ezekre a mérési eredményekre épülnek. A szenzormodul kizárólag adatgyűjtést végez, a döntéshozatal nem itt történik.
- **Vezérlőegység**
Ez a rendszer agya, amely feldolgozza a szenzoradatokat, kiszámítja a sebességet, meghatározza a célsebességhez tartozó eltéréseket, és ennek megfelelően állítja a terhelést. A DR-4, LLR-4 és LLR-5 követelmények közvetlenül ehhez a modulhoz kapcsolódnak, mivel itt történik a sebesség meghatározása, a célértékekkel való összehasonlítás, valamint az ellenállás változtatására vonatkozó parancsok meghozatala.
- **Ellenállás szabályozó egység**
A vezérlőegység által meghatározott parancsok alapján fizikailag növeli vagy csökkenti az ellenállást. Ez lehet motoros fékrendszer, elektromágneses ellenállás vagy más mechanikus megoldás. A modul működése közvetlenül felelős a HLR-2 megvalósításáért, valamint a LLR-6 és LLR-7 követelmények teljesítéséért, amelyek meghatározzák, hogy az ellenállás maximum 10%-os lépésekben változhat.
- **Felhasználói felület**

Ez az a modul, amely a sportoló számára valós idejű visszajelzést jelenít meg (sebesség, terhelés, ismétlésszám). A kijelző frissítésére vonatkozó követelmények (HLR-3, DR-8, LLR-11, LLR-12) miatt ez a blokk biztosítja, hogy a sportoló mindig friss információt kapjon a teljesítményéről. A felhasználói felület csak megjelenítést végez, a számításokat a vezérlőegység végzi.

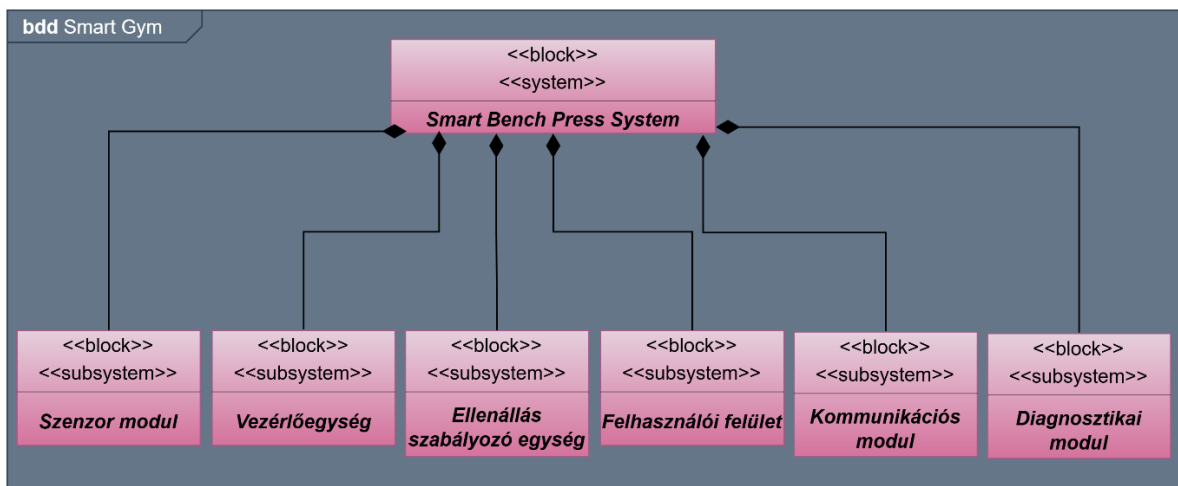
- **Kommunikációs modul**

Ez a modul biztosítja a kapcsolatot a Smart Bench Press és a mobilalkalmazás között (pl. Bluetooth vagy Wi-Fi). Ehhez a modulhoz kapcsolódnak a HLR-4 és DR-6 követelmények, vagyis az edzésadatok továbbítása, a kapcsolatkezelés, valamint a sikeres vagy sikertelen adatátvitel visszajelzése. A kommunikációs modul feladata, hogy megbízható kommunikációt biztosítson.

- **Diagnosztikai modul**

A diagnosztikai adatok tárolásáért és megjelenítéséért felelős blokk. A hibakódok generálása, a hibanapló kezelése, valamint ezek elérhetővé tétele a karbantartó számára (HLR-6, DR-7, LLR-9, LLR-10) mind ehhez a modulhoz tartozik. A rendszer hosszú távú megbízhatóságához elengedhetetlen a hibák naplózása és a karbantartói ellenőrzés lehetősége.

Az alábbi diagram (3. ábra) a Smart Bench Press rendszer fő komponenseinek hierarchiáját és a köztük fennálló egész–rész kapcsolatot mutatja be. A fekete rombusz jelöli, hogy az alrendszerek nem létezhetnek önnálón, csak a rendszer részeként.



3. Block Definition Diagram (saját szerkesztés)

6.2. Internal Block Diagram

Míg a Block Definition Diagram azt rögzíti, hogy milyen alrendszerekből épül fel a rendszer, az Internal Block Diagram ezen alrendszerek konkrét összeköttetéseire fókuszál, hogy melyik modul milyen információt ad át a másiknak, és milyen szerepet tölt be a teljes edzésfolyamatban. [5] [6]

A Smart Bench Press esetében a vezérlés központi eleme a vezérlőegység, amely a szenzormodultól érkező mérési adatokat fogadja, kiszámítja a sebességet, összeveti azt a konfigurációban rögzített célértékekkel, majd ennek megfelelően utasítja az ellenállásszabályzó egységet. Ugyanez a vezérlőegység szolgáltatja a kijelzőn megjelenő információkat, előkészíti az edzésösszesítőt a mobilalkalmazás felé, és továbbítja a diagnosztikai eseményeket a hibanaplózásért felelős modulnak.

A belső struktúra szempontjából a következő portok jelennek meg a SmartBenchPressSystem belsejében:

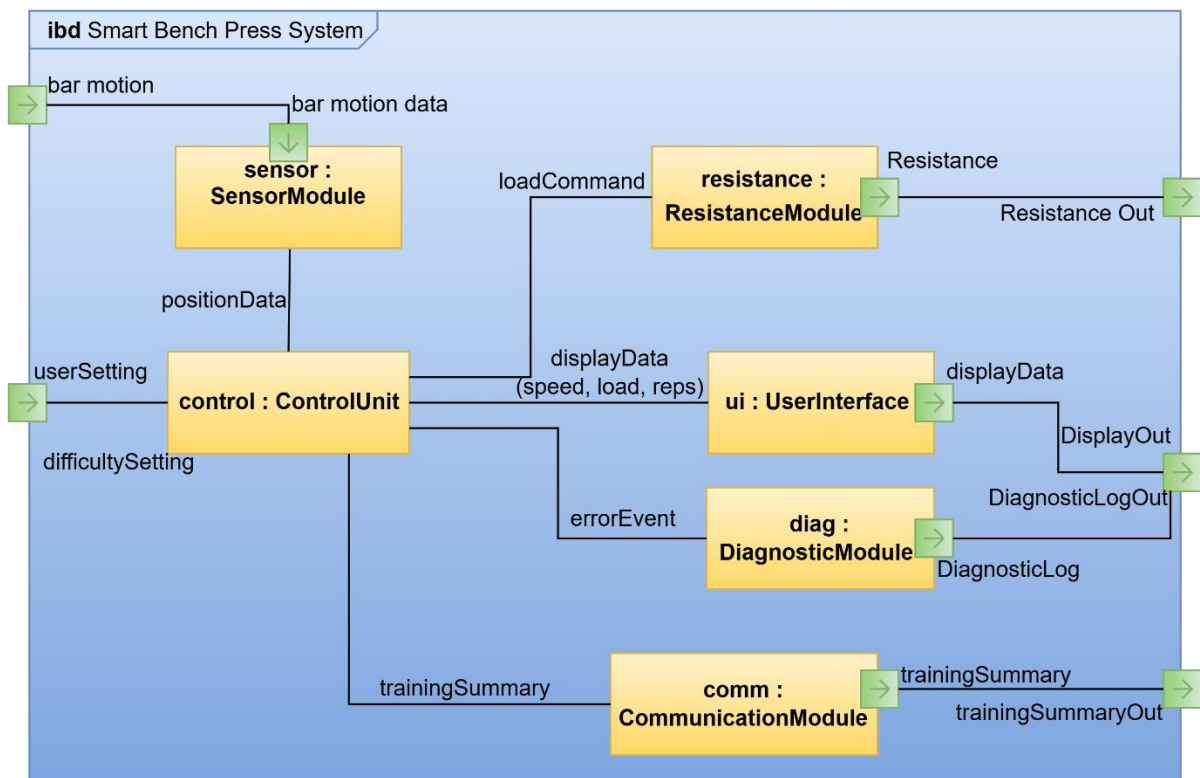
- **sensor : SensorModule** - mérési adatok forrása.
- **control : ControlUnit** - szabályozási logika, sebességszámítás, ellenállás-parancsok.
- **resistance : ResistanceModule** - a vezérlő által meghatározott ellenállás fizikai megvalósítása.
- **ui : UserInterface** - edzésparaméterek valós idejű megjelenítése a sportoló számára.
- **comm : CommunicationModule** - edzésösszesítő továbbítása a mobilalkalmazás irányába.
- **diag : DiagnosticModule** - hibakódok generálása és hibanapló tárolása

Az Internal Block Diagramban a portok közötti kapcsolatok irányított adatáramlásként jelennek meg.

- A **SensorModule** az edzés során a rúd mozgásából származó mért pozíció- és időadatokat továbbítja a **ControlUnit** számára,
- amely ezekből sebességet számít, majd a sebesség és a felhasználói beállítás (nehézségi szint) alapján a **ResistanceModule** felé küld ellenállásmódosító parancsokat.
- A **ResistanceModule** ennek megfelelően módosítja az ellenállást, ezáltal lép kapcsolatba a rendszerhatáron túli fizikai világgal. Mondhatjuk úgy, hogy ezáltal adatot, információt juttat a fizikai világba.
- Szintén a **ControlUnit**-ből származnak a **UserInterface** által megjelenített edzésparaméterek is (aktuális sebesség, ellenállás, ismétlésszám),
- amely a Smart Bench Press System kijelzőjén keresztül juttatja el az információkat a rendszerhatáron túlra.
- Az edzés végén a vezérlőegység előállítja az edzésösszesítőt a **CommunicationModule** számára,
- amelyet tovább küld a rendszerhatáron túli mobilalkalmazás felé.
- Hiba vagy rendellenes működés esetén a vezérlőegység diagnosztikai üzeneteket küld a **DiagnosticModule** számára, amely az eseményeket naplózza
- és igény esetén a kijelzőn keresztül teszi elérhetővé a rendszerhatáron kívül.

Az Internal Block Diagram (4. ábra) célja, hogy szemléletesen megmutassa, hogy a rendszer belső kommunikációja egy jól definiált adatútvonal mentén történik, ahol a ControlUnit a

központi csomópont. Ez a felépítés biztosítja, hogy a követelményekben rögzített automatikus ellenállás szabályozás, valós idejű kijelzés és adatmentés egymással összehangolt módon valósuljon meg.



4. Internal Block Diagram (saját szerkesztés)

6.3 Parametrikus diagram

A parametrikus diagram célja, hogy a Smart Bench Press rendszer működése szempontjából legfontosabb mennyiségek közötti összefüggéseket formálisan is rögzítse. Míg a BDD és az IBD azt mutatja meg, **milyen modulokból áll a rendszer és hogyan kommunikálnak**, addig a parametrikus diagram arra ad választ, hogy **milyen számítások és korlátozások alapján** dönti el a rendszer, mekkora ellenállást kell alkalmazni az egyes ismétlések során. A rendszer működésének és teljesítményének meghatározásához szükséges a kulcsfontosságú fizikai és működésbeli jellemzők azonosítása. Ezeket a mérhető jellemzőket nevezzük **értéktulajdonságoknak (Value Properties)** :

- **position** a rúd mért pozíciója (méterben)
- **timestamp** a mérés időbélyege (ms)
- **speed**
 - **currentSpeed** a rúd aktuális sebessége, a két pozíciókülönbség alapján számolva (m/s)
 - **targetSpeed** a kiválasztott nehézségi szinthez tartozó célsebesség (m/s)
- **error** a célsebesség és a mért sebesség különbsége
- **tolerance** a hibahatár, amin belül a rendszer elfogadható eltérést enged (+/-10%)

- **resistance** ellenállási érték

A rendszer viselkedésének matematikai modellezéséhez és a kényszerek (constraints) elemzéséhez a fenti **értéktulajdonságokból származtatjuk a modell paramétereit.**

A Smart Bench Press esetében a legkritikusabb paraméterek:

- a rúd aktuális pozíciója és az ehhez tartozó időpont,
- az ezekből számított sebesség,
- a nehézségi szinthez tartozó célsebesség,
- a célsebesség és a mért sebesség közötti eltérés,
- a szükséges ellenállásváltozás, amelyet legfeljebb 10%-os lépésekben lehet módosítani.

A diagram három fő **constraint block** köré szerveződik, amelyek a fenti kapcsolatokat írják le:

1. VelocityCalculate

Ez a blokk a rúd sebességének kiszámítását végzi a pozíció és az idő alapján.

Paraméterek

Paraméter	Leírás
x1	rúdpozíció az előző mintavételezéskor
x2	rúdpozíció a jelenlegi mintavételezéskor
t1	időbélyeg az előző mintavételezéskor
t2	időbélyeg a jelenlegi mintavételezéskor
v	kiszámított sebesség

Korlátozás (constraint)

- $v = (x2 - x1) / (t2 - t1)$

Ez a blokk a LLR-4 (Sebességszámítás) követelménynek felel meg.

2. SpeedErrorAndTolerance

Ez a blokk a célsebességhez viszonyított eltérést és a hibahatárt írja le.

Paraméterek

Paraméter	Leírás
v_target	a nehézségi szinthez tartozó célsebesség
v	mért sebesség (a VelocityCalculate eredménye)
error	a sebességhiba
tol	engedélyezett hibahatár +/-10%

Korlátozások (constraint)

- $error = v_target - v$
- $abs(error) \leq tol$

Ez a blokk a DR-2 és LLR-2, LLR-5 követelményekhez kapcsolódik, mivel meghatározza, milyen eltérés engedhető meg a célsebességhez képest.

3. ResistanceUpdate

Ez a blokk azt fejezi ki, hogy a célsebességtől függően hogyan módosulhat az ellenállás.

Paraméterek

Paraméter	Leírás
R_prev	előző ciklusban alkalmazott ellenállás
R_new	új ellenállás
v_target	célsebesség
v	mért sebesség (a VelocityCalculate eredménye)

Korlátozások

- if $0.9 \leq v_target/v \leq 1.1$ then $R_new = R_prev * v_target/v$
- if $v_target/v < 0.9$ then $R_new = R_prev * 0.9$
- if $v_target/v > 0.9$ then $R_new = R_prev * 1.1$
- $R_prev \geq 20 \text{ Nm}$ $R_new \geq 20 \text{ Nm}$

A részletes vezérlési logika (pl. növelés vagy csökkentés iránya) a ControlUnit implementációjában jelenik meg, a parametrikus diagram itt elsősorban a változás mértékét korlátozza, összhangban az LLR-6 és LLR-7 követelményekkel.

A parametrikus diagram (5. ábra) a SmartBenchPressSystem blokk belsejében elhelyezett értéktulajdonságokat köti össze a fenti constraint blockok paramétereivel.

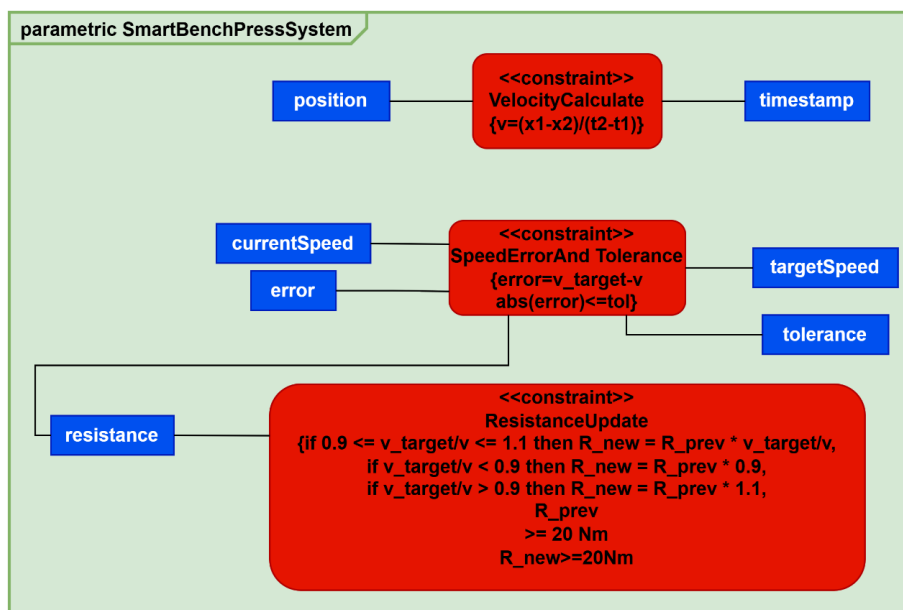
- a **SensorModule** rendelkezik position és timestamp értéktulajdonságokkal, amelyek a VelocityCalculate blokk x1, x2, t1, t2 paramétereire csatlakoznak.
- a **ControlUnit** tartalmazza a currentSpeed, targetSpeed, error, tolerance és resistance értéktulajdonságokat, amelyek a SpeedErrorAndTolerance és a ResistanceUpdate blokkok paramétereire kapcsolódnak.
- a **ResistanceModule** resistance értéktulajdonsága a R_new paraméterrel áll kapcsolatban.

A diagram így szemléletesen megmutatja, hogy a rendszer:

1. a szenzormodul által szolgáltatott pozíció–idő adatokból sebességet számít,
2. ezt a sebességet összeveti a nehézségi szinthez tartozó célsebességgel,
3. majd az eltérés és a megengedett hibahatár figyelembevételével úgy módosítja az ellenállást, hogy annak változása ne haladhassa meg a korábbi érték 10%-át.

Ezzel a parametrikus diagram közvetlenül összeköti a követelményeket (LLR-4, LLR-5, LLR-6, LLR-7) a rendszer numerikus modelljével.

[7]



5. Parametric Diagram (saját szerkesztés)

7. Rendszer viselkedése

Ebben a fejezetben azt mutatom be, hogy **időben hogyan viselkedik a rendszer**: milyen lépések egymásutánja vezet egy-egy funkció megvalósításához, hogyan zajlik a modulok közti üzenetváltás, illetve milyen állapotokon megy keresztül a rendszer.

A fejezetben háromféle viselkedésmódot alkalmazok:

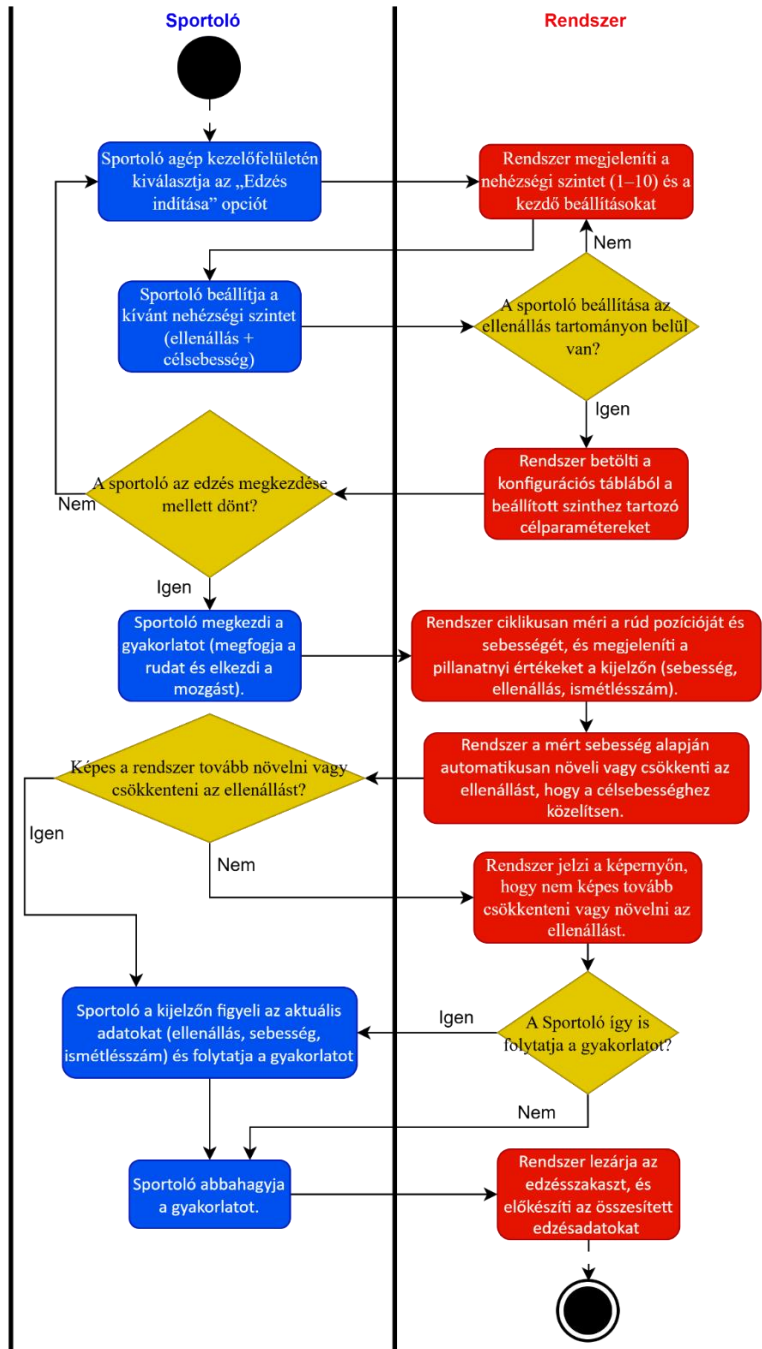
1. **Aktivitási diagramok**, amelyek egy-egy használati eset lépéseit folyamatábrászerűen ábrázolják.
2. **Szekvencia diagram**, amely az aktorok és alrendszerek közötti üzenetváltást mutatja.
3. **Állapotgép diagram**, amely a Smart Bench Press főbb működési állapotait és az állapotváltásokat rögzíti.

7.1 Aktivitási diagramok

Az aktivitási diagramokat a korábban definiált **use case forgatókönyvekre** alapozva készítem el. Míg a forgatókönyv szövegesen sorolja fel a lépéseket, addig az aktivitási diagram ezeket átlátható ábraként jeleníti meg. Az aktivitási diagramokat **swimlane-ekre** osztottam, hogy elkülönüljön, melyik lépést hajtja végre az adott aktor és melyiket a Smart Bench Press rendszer.

7.1.1 Aktivitási diagram - UC1: Edzés végrehajtása

Ez a diagram (6. ábra) az UC1 Edzés végrehajtása használati eset lépéseit jeleníti meg. A swimlane-ek elkülönítik a sportoló és a Smart Bench Press rendszer felelősségi köreit. Jól látható, hogy az edzés egy ciklikus folyamat. A sportoló általi rúd mozgása alapján a rendszer folyamatosan méri a pozíciót és sebességet, ellenőrzi a hibát a célsebességhez képest, szükség esetén módosítja az ellenállást, és valós időben frissíti a kijelzett edzésparamétereket. A diagram tartalmazza az érvénytelen beállításra, az edzés megkezdése előtti megszakításra és a normál befejezésre vonatkozó alternatív ágakat is.

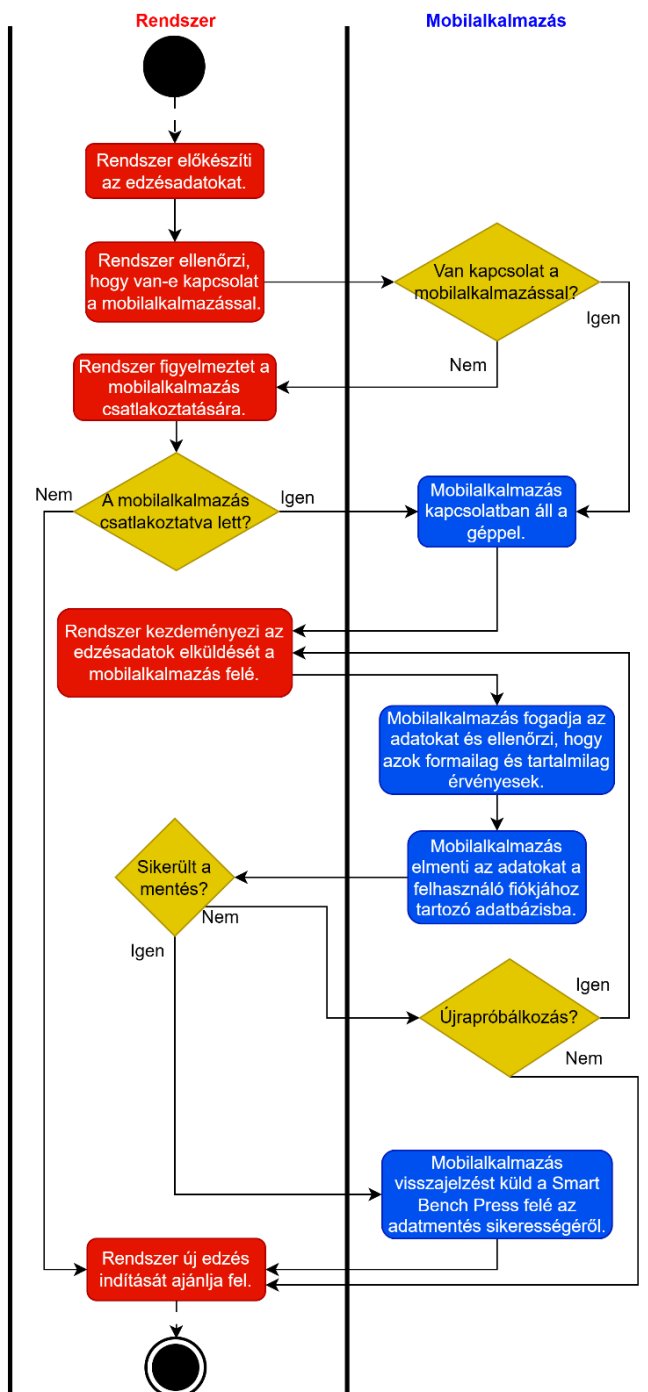


6. Activity Diagram UC1 (saját szerkesztés)

7.1.2 Aktivitási diagram – UC2: Edzésadatok mentése

Ez az aktivitási diagram (7. ábra) az **UC2 Edzésadatok mentése** használati eset folyamatát szemlélteti. A swimlane-ek elkülönítik a Smart Bench Press rendszer és a mobilalkalmazás felelősségi köreit. A diagram bemutatja, hogyan kerülnek az edzés során összegyűjtött adatok továbbításra és mentésre a felhasználói fiókhhoz tartozó adatbázisban. A folyamat magában foglalja az adatátviteli kapcsolat ellenőrzését, az edzésadatok átküldését, a mobilalkalmazás általi érvényesség-ellenőrzést, valamint a mentés eredményéről szóló visszajelzést. A diagram alternatív ágakon kezeli a kapcsolat megszakadását és a sikertelen mentési eseteket is, ezzel

biztosítva, hogy az edzésadatok ne vesszenek el, és a felhasználó minden esetben értesüljön az állapotról.

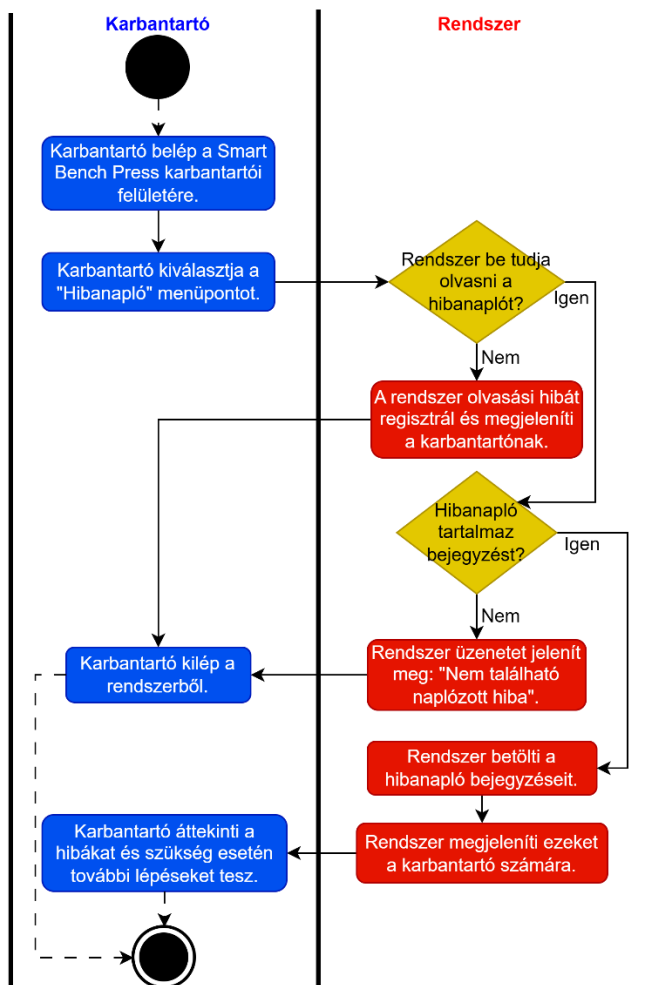


7. Activity Diagram UC2 (saját szerkesztés)

7.1.3 Aktivitási diagram – UC3: Diagnosztikai adatok lekérdezése

Ez az aktivitási diagram (8. ábra) az UC3 Diagnosztikai adatok lekérdezése használati eset megvalósítását írja le. A folyamat bemutatja a karbantartó és a Smart Bench Press rendszer közötti interakciót karbantartói módban. Jól látható, hogy a karbantartó a diagnosztikai felületen keresztül lekérdezheti a hibanaplót, míg a rendszer betölti és megjeleníti a rögzített

hibabejegyzéseket. A diagram alternatív ágakon kezeli azt az esetet is, amikor a hibanapló üres vagy nem elérhető, biztosítva ezzel, hogy a karbantartó minden helyzetben egyértelmű visszajelzést kapjon a rendszer állapotáról.



8. Activity Diagram UC3 (saját szerkesztés)

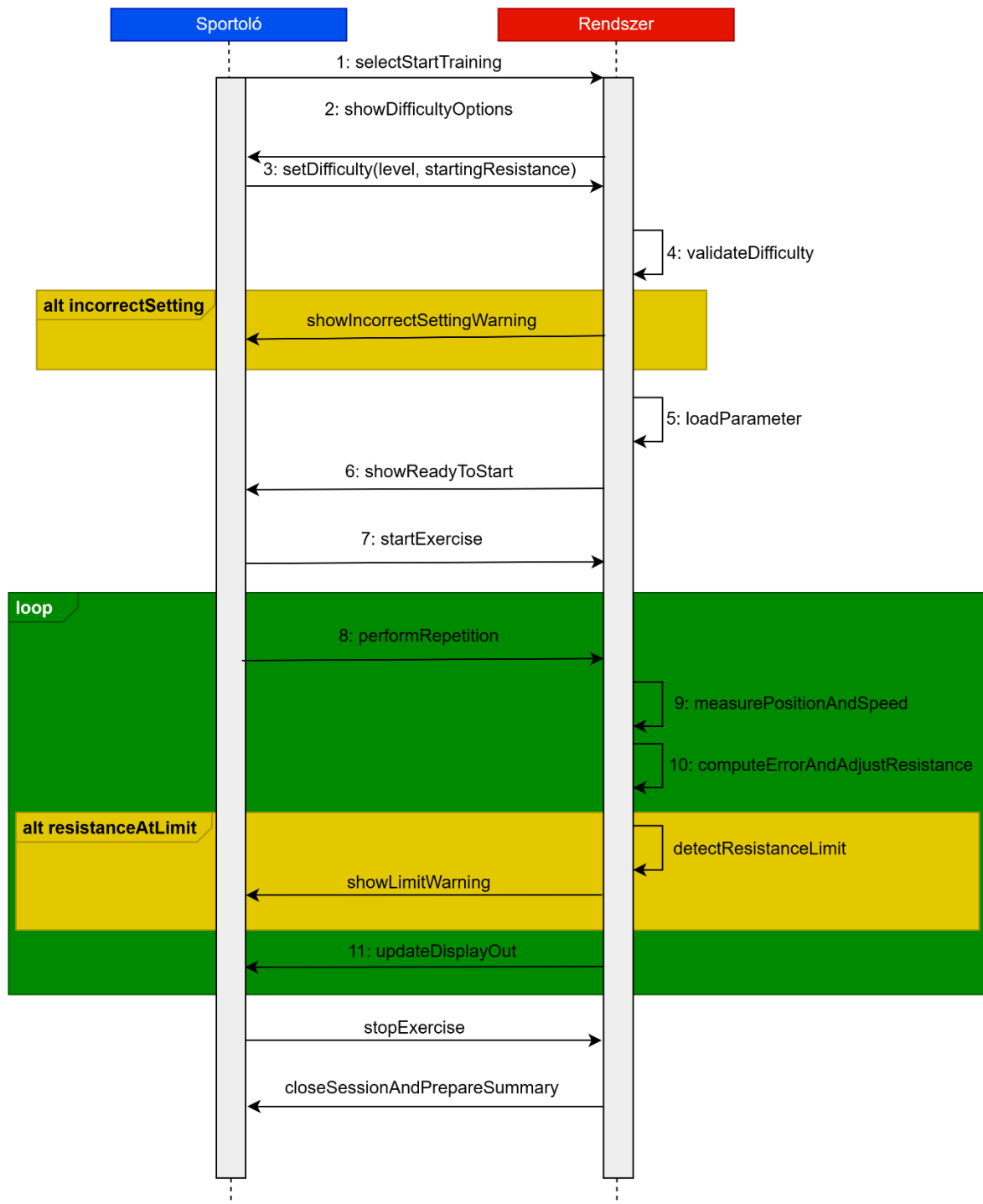
7.2 Szekvenciadiagramok

Míg az aktivitási diagram az egyes lépések folyamatát és elágazásait hangsúlyozza, addig a szekvenciadiagram azt ábrázolja, hogy **melyik szereplő vagy alrendszer kivel kommunikál, és milyen üzeneteket cserélnek** egymással az idő előrehaladtával. A Smart Bench Press esetében a szekvenciadiagramok a korábban azonosított használati esetek (UC1–UC3) fő forgatókönyveit követik, de már nem csak az aktorokra, hanem a belső modulokra (vezérlőegység, szenzormodul, ellenállás-szabályozó egység, felhasználói felület, kommunikációs modul) is fókuszálnak. [8]

7.2.1 UC1 Edzés végrehajtása szekvenciadiagram

A következő szekvenciadiagram (9. ábra) az **UC1 Edzés végrehajtása** használati eset időbeli lefolyását mutatja be a Sportoló és a Smart Bench Press System között. A sportoló elindítja az edzést, beállítja a nehézségi szintet, a rendszer betölti a konfigurációs paramétereket, majd

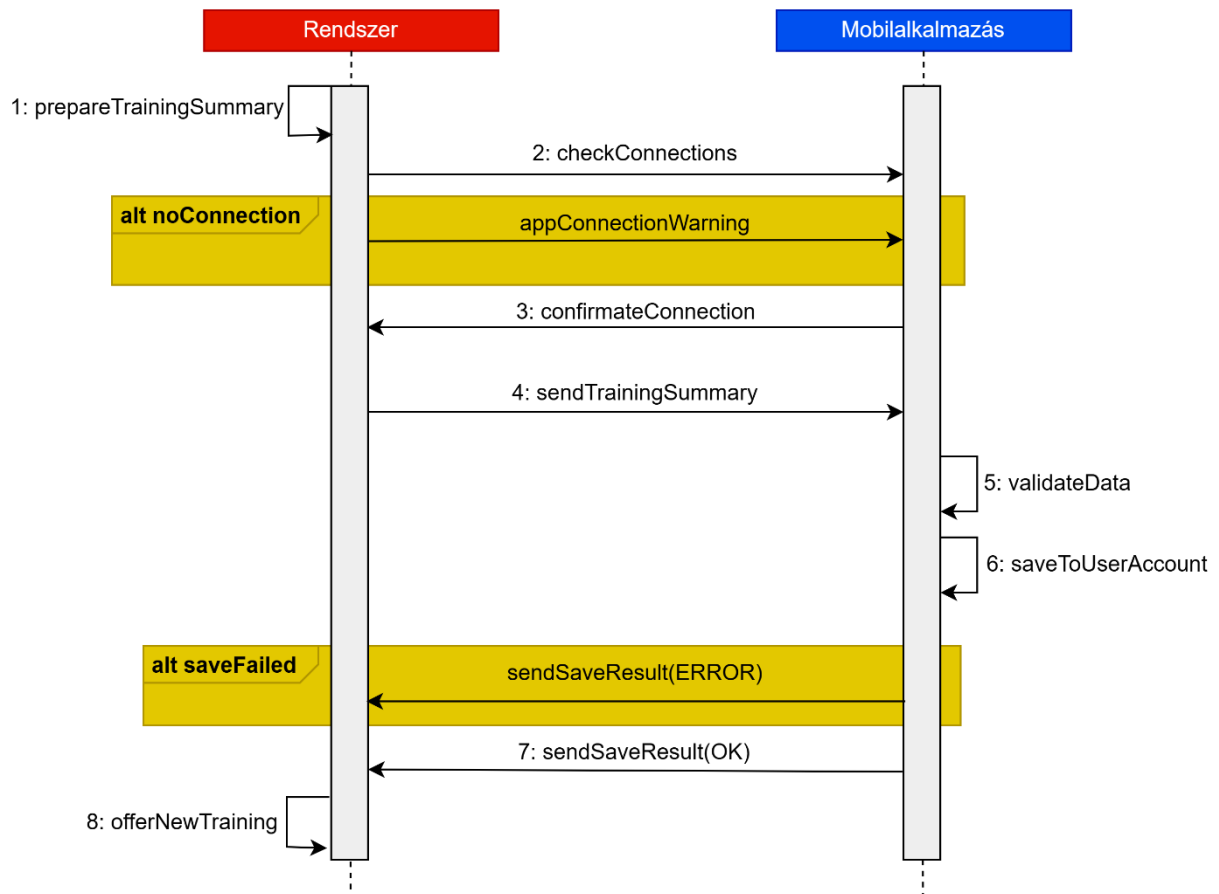
egy ciklusban méri a rúd mozgását, kiszámítja a sebességet, korigálja az ellenállást és frissíti a kijelzett edzésadatokat. A loop fragment a gyakorlat ismétlődő részét írja le, míg az alt fragmentek az érvénytelen nehézségi szint és az ellenállási küszöb elérésére vonatkozó alternatív viselkedéseket mutatják. A szekvencia diagram a sportoló által kezdeményezett edzésleállítással és az edzésösszesítő előkészítésével zárul.



9. Sequence Diagram UC1 (saját szerkesztés)

7.2.2 UC2 Edzésadatok mentése

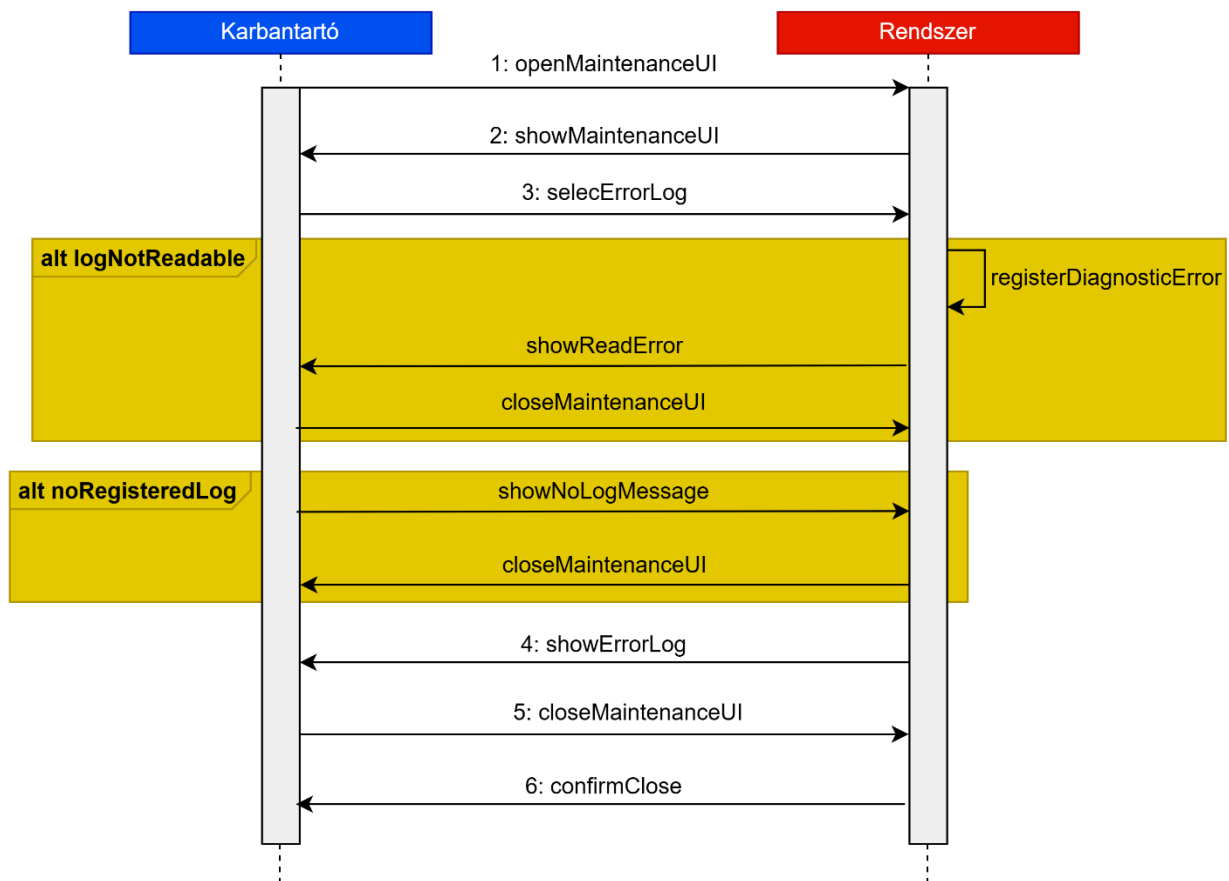
Ez a szekvenciadiagram (10. ábra) az **UC2 Edzésadatok mentése** használati eset üzenetváltásait írja le a Smart Bench Press Rendszer és a Mobilalkalmazás között. A Rendszer előkészíti az edzésösszesítőt, ellenőrzi, hogy van-e aktív kapcsolat a mobilalkalmazással, majd sikeres kapcsolat esetén elküldi az edzésadatokat. A mobilalkalmazás ellenőrzi az adatok érvényességét, elmenti azokat a felhasználói fiókhoz tartozó adatbázisba, és visszajelzést küld a mentés sikerességéről vagy sikertelenségéről. A diagram alt fragmentekkel kezeli a kapcsolat hiányát és a sikertelen mentést, a folyamat végén pedig a Rendszer új edzés indítását ajánlja fel.



10. Sequence Diagram UC2 (saját szerkesztés)

7.2.3 UC3 Diagnosztikai adatok lekérdezése

Ez a szekvenciadiagram (11. ábra) az **UC3 Diagnosztikai adatok lekérdezése** használati esetet mutatja be a karbantartó és a rendszer közötti üzenetváltások szintjén. A karbantartó belép a karbantartói felületre, kiválasztja a hibanapló megtekintését, a rendszer pedig megpróbálja beolvasni a naplót. Az alt blokkok jelzik a lehetséges elágazásokat. Ha a rendszer a hibanaplót nem tudja beolvasni, a rendszer diagnosztikai hibát regisztrál és hibaüzenetet jelenít meg. Ha a napló olvasható, akkor vagy a naplózott hibák listája jelenik meg, vagy egy üzenet arról, hogy nem található bejegyzés. A diagram így szemlélteti, hogyan támogatja a rendszer a karbantartót a hibakeresésben, és hogyan kezeli a diagnosztikai alrendszer saját hibáit is.



11. Sequence Diagram UC3 (saját szerkesztés)

7.3 Állapotgép diagram

Az állapotgép-diagram (12. ábra) a Smart Bench Press rendszer fő üzemállapotait és azok közötti átmeneteket mutatja be. A diagram elsősorban az UC1 Edzés végrehajtása és UC2 Edzésadatok mentése használati esetekhez kapcsolódik, kiegészítve egy karbantartói és egy hibaállapottal. Az állapotok azt írják le, hogy a rendszer éppen milyen működési fázisban van, az átmenetek pedig a felhasználói események (gombnyomás, rúd megmozdítása) vagy belső feltételek (ellenállás-határ elérése, mentés sikeressége) teljesülésekor következnek be.

A rendszer állapotai a következők:

1. Idle

A rendszer be van kapcsolva, de nincs aktív edzésszakasz.

2. SettingDifficulty

A sportoló beállítja a nehézségi szintet (ellenállás + célsebesség).

3. WaitingForStart

A nehézségi szint már be van állítva, a gép várja, hogy a sportoló ténylegesen megkezdje a gyakorlatot.

4. TrainingActive

Ciklikus mérés, sebességszámítás, hibaszámítás, ellenállás állítása, kijelző frissítése.

5. ResistanceLimitWarning

A rendszer már nem tudja tovább növelni/csökkenteni az ellenállást.

6. PrparingSummary

A rendszer összesíti az edzésadatokat.

7. ReadyToSend

Az összesített adat készen áll elküldésre a mobilalkalmazás felé.

8. SendingData

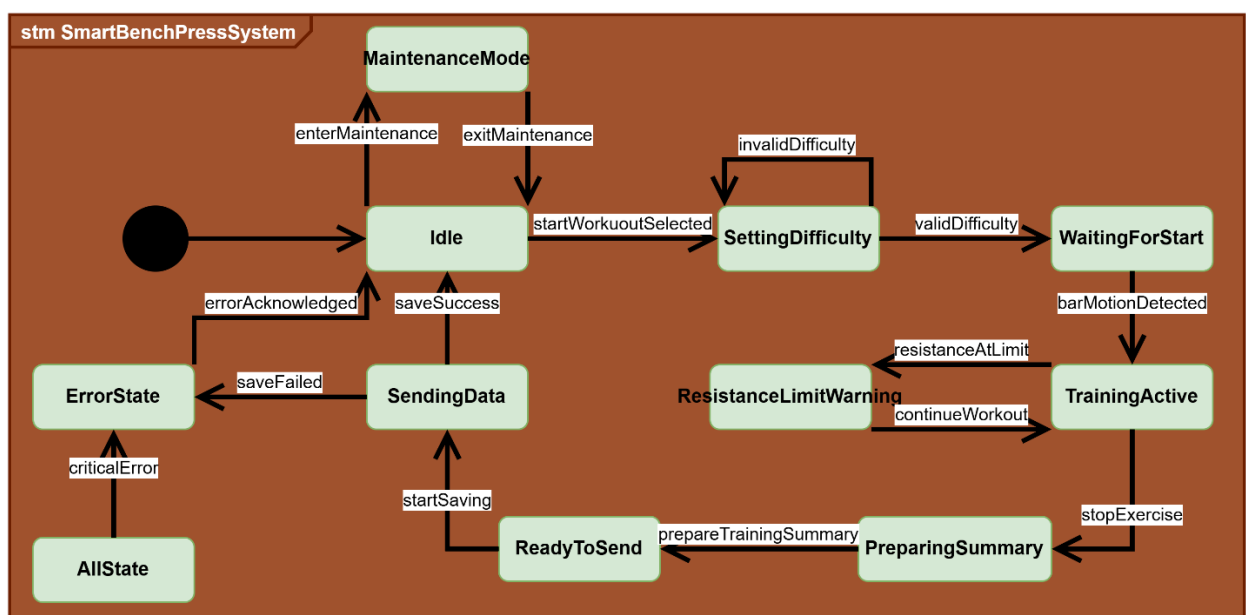
Rendszer elküldi az edzésadatokat a mobilalkalmazásnak, és vár a válasza.

9. MaintenanceMode

Karbantartó belépett a karbantartói felületre, hibanaplót néz stb.

10. ErrorState

Kritikus hiba esetén.



12. StateMachine Diagram (saját szerkesztés)

8. Összefoglalás

A beadandó egy Smart Bench Press rendszer SysML alapú modellezését mutatta be. A munka során meghatároztam a rendszer célját és a megoldandó problémákat, azonosítottam az érintetteket, majd hierarchikusan felépített követelményrendszert (HLR, DR, LLR) hoztam létre.

Ezután a rendszerhatár kijelölésével, aktorok azonosításával, use case-ek azonosításával és a use case forgatókönyvek megírásával azonosítottam a használati eseteket.

A strukturális modellekkel (BDD, IBD) feltártam a rendszer belső felépítését és adatáramlását, míg a parametrikus diagrammal leírtam az ellenállásszabályozás matematikai összefüggéseit.

A viselkedési diagramok (aktivitási, szekvencia és állapotgép diagramok) időbeli nézőpontból is bemutatták a rendszer működését. A beadandó célja nem egy konkrét implementáció volt,

hanem annak demonstrálása, hogy a SysML hatékony eszköz egy kiberfizikai rendszer átlátható és következetes modellezésére.

Ábrajegyzék

1. Requirement Diagram (saját szerkesztés)	8
2. Use Case Diagram (saját szerkesztés)	10
3. Block Definition Diagram (saját szerkesztés)	14
4. Internal Block Diagram (saját szerkesztés)	16
5. Parametric Diagram (saját szerkesztés)	19
6. Activity Diagram UC1 (saját szerkesztés)	20
7. Activity Diagram UC2 (saját szerkesztés)	21
8. Activity Diagram UC3 (saját szerkesztés)	22
9. Sequence Diagram UC1 (saját szerkesztés)	23
10. Sequence Diagram UC2 (saját szerkesztés).....	24
11. Sequence Diagram UC3 (saját szerkesztés).....	25
12. StateMachine Diagram (saját szerkesztés)	26

Irodalomjegyzék

[1] Ian Sommerville, *Software Engineering Ninth Edition 4.5 fejezet* [Hozzáférés dátuma: 2025.12.19] [Software Engineering, 9th Edition](#)

[2] Hajdu Csaba, *Ipar 4.0 technológiák (GKLB_INTM087) tárgy, Rendszertervezés, SysML 2. SysML előadás, Követelmények szintjei (DO-178C alapján) dia* [Hozzáférés dátuma: 2025. 11. 28] [Distributed cognitive architecture for robotic and cyber-physical systems](#)

[3] Ian Sommerville, *Software Engineering Ninth Edition 4.5.4 fejezet* [Hozzáférés dátuma: 2025.12.19] [Software Engineering, 9th Edition](#)

[4] Hajdu Csaba, *Ipar 4.0 technológiák (GKLB_INTM087) tárgy, Rendszertervezés, SysML 2. SysML előadás, Modellelés SysML segítségével dia* [Hozzáférés dátuma: 2025. 11. 28] [Distributed cognitive architecture for robotic and cyber-physical systems](#)

[5] SparxSystem, *Block Definition Diagram* [Hozzáférés dátuma: 2025. 11.30] [Block Definition Diagram | Enterprise Architect User Guide](#)

[6] SparxSystem, *Internal Block Diagram* [Hozzáférés dátuma: 2025. 11.30] [Internal Block Diagram | Enterprise Architect User Guide](#)

[7] SparxSystem, *Parametric Diagram* [Hozzáférés dátuma: 2025. 11.30] [Parametric Diagram | Enterprise Architect User Guide](#)

[8] Ian Sommerville, *Software Engineering Ninth Edition 5.2.2 fejezet* [Hozzáférés dátuma: 2025.12.19] [Software Engineering, 9th Edition](#)

