**SRPLS, Student RFID Phone Locker System**

**Vizsgaremek dokumentáció**

A SRPLS (Student RFID Phone Locker System) egy olyan RFID alapú rendszer, amely lehetővé teszi az iskolák számára, hogy automatikusan kezeljék a diákok telefonjainak tárolását. A diákok minden nap reggel a telefonjukat egy szekrénybe helyezik, amely csak az ő egyedi RFID kártyájukkal nyitható ki. A rendszer biztonságos, automatizált, és lehetőséget nyújt a tanároknak, adminisztrátoroknak, hogy szükség esetén hozzáférést adjanak a diákoknak. A rendszer magában foglalja a hardveres (RFID olvasó és záró mechanizmus) és szoftveres (adatbázis, webes felület, adminisztrációs rendszer) részeket.

# **Funkcionalitás**

1. **RFID alapú azonosítás:**

A diákok RFID kártyákat kapnak, amelyek az egyedi szekrényekhez kapcsolódnak. A kártya segítségével a szekrény nyitható reggelente, és az iskolai nap végén.

1. **Automatikus zárás és nyitás:**

A szekrények az órarend alapján automatikusan zárnak, amikor elkezdődik az

iskola, és nyílnak, amikor véget ér a tanítási nap.

1. **Tanári jóváhagyás:**

Egyes helyzetekben (pl. egy diák kihagy egy órát) tanári jóváhagyás szükséges a szekrény nyitásához, amit az adminisztrátorok végezhetnek el a webes felületen keresztül.

1. **Valós idejű nyitás/zárás naplózás:**

Minden szekrény nyitás/zárás időpontja és az esemény engedélyezője naplózva van

az adatbázisban.

# **Adatbázis Struktúra**

Az adatbázis a rendszer alapját képezi, amely a diákok, szekrények, hozzáférési történetek, tantárgyak és adminisztrátorok adatait tárolja.

1. students (Diákok)
   1. student\_id: A diákok egyedi azonosítója.
   2. full\_name: A diák teljes neve.
   3. class: A diák osztálya.
   4. rfid\_tag: A diák RFID kártyájának egyedi azonosítója.
2. locker\_access\_history (Hozzáférési Napló)
   1. access\_id: Egyedi azonosító minden hozzáférési eseményhez.
   2. rfid\_tag: Az RFID azonosító, amellyel a szekrényhez hozzáfértek.
   3. locker\_id: A hozzáfért szekrény azonosítója.
   4. access\_time: A hozzáférés időpontja.
   5. approved\_by: Az adminisztrátor azonosítója, aki jóváhagyta a hozzáférést (ha szükséges volt).
3. locker\_relationships (Diák és szekrény kapcsolatok)
   1. relationship\_id: Egyedi azonosító minden diák és szekrény kapcsolat számára.
   2. rfid\_tag: A diák RFID azonosítója.
   3. locker\_id: A diákhoz rendelt szekrény azonosítója.
4. subjects (Tantárgyak)
   1. subject\_id: Egyedi azonosító minden tantárgyhoz.
   2. teacher\_name: A tantárgyért felelős tanár neve.
5. groups (Csoportok)
   1. group\_id: Egyedi azonosító minden csoporthoz.
   2. group\_name: A csoport neve (pl. osztályok).
6. student\_groups (Diák-csoport kapcsolatok)
   1. student\_group\_id: Egyedi azonosító minden diák-csoport kapcsolat számára.
   2. student\_id: A diák azonosítója.
   3. group\_id: A csoport azonosítója, amelyhez a diák tartozik.
7. timetables (Órarend)
   1. timetable\_id: Egyedi azonosító az órarendhez.
   2. group\_id: Az órarendhez kapcsolt csoport azonosítója.
   3. subject\_id: A tantárgy azonosítója.
   4. day\_of\_week: Az óra napja (Hétfő-Péntek).
   5. start\_time: Az óra kezdete.
   6. end\_time: Az óra vége.
8. admins (Adminisztrátorok)
   1. admin\_id: Egyedi azonosító minden adminisztrátorhoz.
   2. full\_name: Az adminisztrátor neve.
   3. position: Az adminisztrátor pozíciója (pl. igazgató, rendszergazda).

# **Webes Felület**

A webes adminisztrációs felület lehetővé teszi a tanárok és adminisztrátorok számára:

* Diákok hozzáférésének figyelését és naplózását.
* Az órarend kezelhetőségét és automatikus szekrény-zárások beállítását az iskolai nap alapján.

# **API Funkciók**

Az SRPLS rendszer kommunikációs rétege egy REST API-t használ, amely lehetővé teszi

a külső rendszerek, az Arduino eszközök és a backend közötti adatcserét.

* Szekrény nyitás/zárás API: Az RFID kártyák beolvasásakor a rendszer ellenőrzi,

hogy a szekrény nyitható-e, majd frissíti az adatbázisban a szekrény állapotát.

* Hozzáférési napló API: Minden szekrény nyitás és zárás eseményét rögzíti az API, beleértve a diák azonosítóját és az esemény időpontját.

# **Hardver és Szoftver Integráció**

A hardveres komponensek (RFID olvasók, zárak) közvetlenül kapcsolódnak a szoftveres rendszerhez, és az adatokat egy MySQL adatbázisba küldik, ahol naplózzák a hozzáférési eseményeket. Az Arduino UNO vezérli a zárak nyitását és zárását, és serial kommunikáción keresztül kapja az RFID kártya adatait.

**Hardveres megvalósítás:**

* Arduino UNO, ezen futna a szoftver amely kezli a szekrényeket
* RFID tag, a diákok a tag-ekket letudják csippantani az RFID leolvasónál
* RFID leolvasó, beolvassa az RFID tag-eket, és a hozzárendelt szekrényt kinyitja
* Szolenoid zárak Backend:
* NodeJs
* Python Frontend:
* HTML,CSS
* JS
* Bootstrap

# **Fejlesztési Lehetőségek**

* ASC órarend: A rendszerünkbe integrálnánk az ASC órarendkészítőt, így egy gombnyomással feltölthető lenne az órarend.
* Áramellátás és internetelérés hiányában a szekrények manuális nyitása és zárása.
* Hivatalos helyettesítések és szívességi helyettesítések hozzáadása és kezelése.
* További integrációk: A jövőben lehetőség van más iskolai rendszerekkel való

integrációra, például iskolai beléptető vagy étkezdei rendszerével, illetve más tárolókkal.