

**Fejlesztői dokumentáció**

**Fejlesztés célja1**

Páciens felület1.1

Orvosi felület1.2

Felület tervek2

Felhasznált technológiák3

Reszoponzív webes és mobil alkalmazáshoz 3.1

Asztali alkalmazáshoz 3.2

A backend szerverhez 3.3

Késítéshez használt programok 3.4

Adatszerkezetek4

Kódolási konvenciók5

Könyvtár szerkezet src-n belül:5.1

Könyvtár szerkezet src-n belül:5.2

Végpontok6

Álltalános működés7

Osztályok8

Fejlesztői környezetek9

1. **Fejlesztés célja:**

**1.1 Páciens felület**

A fejlesztés célja, hogy egy olyan felhasználóbarát egészségügyi felhő alkalmazást hozzunk létre, amely segíti a pácienseket nyomon követni a vérvételi eredményeiket. Tudatja a felhasználókat arról, hogy melyik vérvételi komponens miért fontos. A backendet egy Node.js backend szerver biztositja az adatbázist pedig egy MariaDB szerver. Frontend react.js-ben készült single page, responsive webalkalmzás. Telefonon és asztali üzemmódban egyaránt működőképes. A pácienseknek bejelentkezés után lehetősége van megtekinteni a vérvételi adatait ami vizualizálva jelenik meg számára hasznos információkat mutatva arról.

**1.2 Orvosi felület**

Ugyanakkor a fejlesztésnek az is célja volt, hogy ne csak egy pácienseknek szánt alkalmazás készüljön el hanem egy orvosi felület is amely kiegészitve a felhasználói részt megkönnyíti az orvosok dolgát egy egy beteg adatainak a kezelése során. Ezzel az egész rendszert leegyszerűsítve. A backendet egy Node.js backend szerver biztositja az adatbázist pedig egy MariaDB szerver. Frontend electronban és react.js-ben készült asztali alkamazásnak. Az alkalmazás a minimum méretekig teljesen responsivan működik. Az orvosnak a bejelentkezése után lehetősége van megtekinteni a páciensek listáját abban keresni tud valamint részletes vérvételi adatokat megtekinteni. Ezeket módositani,törölni,vagy újat felvenni.

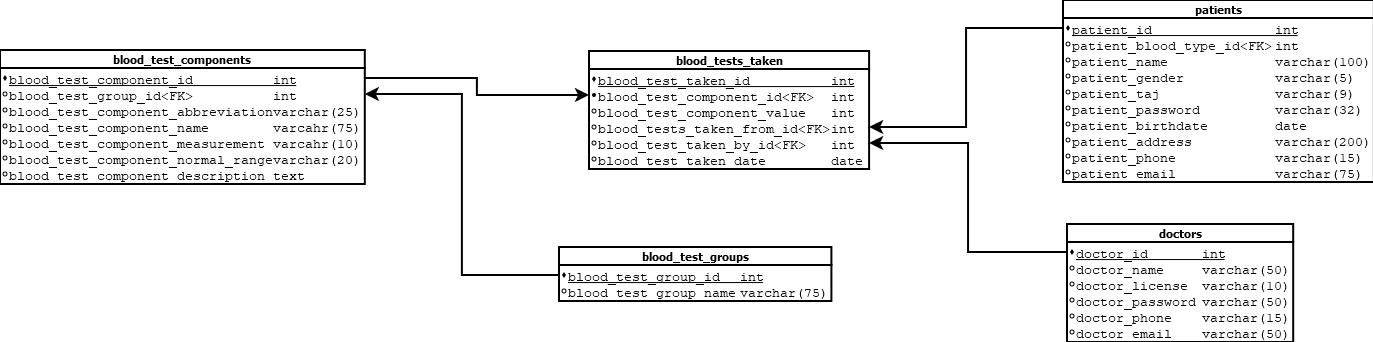
1. **Felülettervek:**
2. **Felhasznált technológiák:**

**3.1** Reszoponzív webes és mobil alkalmazáshoz: React.js, Maridb server(adatbázis kiszolgáló)

**3.3** Asztali alkalmazáshoz: Electron és React.js, Maridb server(adatbázis kiszolgáló)

**3.3** A backend szerverhez: Node.js, Express, Express-session, Cors, MySql

* 1. Késítéshez használt programok: dia, Visual studio code, Isomnia, Mariadb server, phpmyadmin

**4.Adatszerkezetek:**

**5.Kódolási konvenciók**

A fejlesztés során végig git verzió követőt használtunk.

**5.1** **Könyvtár szerkezet:**

* database
* desktop-app
* doc
* responsive-app
* server

**5.2 Könyvtár szerkezet src-n belül:**

* assets
* components
* pages
* utils

A databse tartalmazza az adatbázis importálásához szükséges sql filet. A desktop-app tartalmazza az asztali alkalmazást. A doc könyvtárban a felület tervek és a dokumentációk találhatók a programról. A responsive-app könyvtárban található a webalkalmazás. A server könyvtárban a backend server.

1. Algoritmusok
2. Tesztelés
3. Ismert hibák
4. Készítők:
   1. Oláh János Gergely
   2. Hegedűs György