

# IMS spektrometer

## Návrh

## Contents

1. Úvod .....	3
1.1. Účel.....	3
1.2. Rozsah.....	3
1.3. Definície a skratky .....	3
2. Špecifikácia vonkajších interfejsov .....	3
2.1. Spektrometer .....	3
2.2. ShieldBuddy TC275.....	3
2.3. Meranie iónovej mobility .....	3
3. Dátový model .....	4
3.1. Formáty súborov .....	4
3.2. Komunikačné protokoly .....	4
3.2.1. Sériový port RS232 .....	4
4. Návrh používateľského rozhrania.....	5
4.1. Názov projektu .....	5
4.2. Nastavenia aplikácie.....	5
4.3. Nastavenie Mobility .....	6
4.4. Sekcia grafu .....	7
5. Návrh implementácie .....	9
5.1. Class diagram.....	9
5.2. Stavový diagram - Hlavný graf .....	9
5.3. Diagram komponentov.....	10

## 1. Úvod

### 1.1. Účel

Tento dokument slúži ako dôkladný technický návrh spôsobu implementácie informačného systému pre IMS spektrometer. Je určený primárne pre vývojový tím ako referenčný dokument, ale aj pre zadávateľa na kontrolu.

### 1.2. Rozsah

Informačný systém opísaný v tomto dokumente bude slúžiť na spracovanie údajov z IMS spektrometra. Jeho cieľom je zabezpečiť spracovanie nameraných údajov v podobe grafov. Jeho pridaná hodnota spočíva hlavne v zaznamenaní, sprehľadnení a zhromaždení údajov na jednom mieste.

### 1.3. Definície a skratky

Už uvedené v katalógu požiadaviek

## 2. Špecifikácia vonkajších interfejsov

### 2.1. Spektrometer

Spektrometer slúži na skúmanie prvkového chemického zloženia látky na základe merania vlnovej dĺžky odrazeného svetla od danej látky.

### 2.2. ShieldBuddy TC275

Shieldbuddy TC275 bude riešiť komunikáciu systému so spektrometrom. Bude slúžiť na nastavenie spektrometra podľa požadovaných parametrov zadanych do systému a spustenie merania. Taktiež bude následne čítať namerané hodnoty zo spektrometra a posilať ich do systému. Tieto prvky už boli funkčné v predchádzajúcom systéme, tak sme ich nemenili.

### 2.3. Meranie iónovej mobility

Model	ShieldBuddy TC275
Signál output	5V
Maximálny čas merania	20 milisekúnd
Hustota nameraných bodov (sampling)	1 - 40 (mikrosekundy)
Iónové zrážanie (gate)	1 – 20 (mikrosekundy)

## 3. Dátový model

### 3.1. Formáty súborov

Bez rozdielu na obsah údajov, všetky formáty ukladania budú v CSV (hodnoty oddelené čiarkou). Pre grafy, CSV súbor bude obsahovať riadky zodpovedajúce jednotlivým záznamom rozdelené pomocou oddeľovacieho znaku.

Budú obsahovať presne dva stĺpce pri agregovaných údajoch, kde prvý stĺpec je čas v mikrosekundách a druhý je zaznamenaný signál v bode. Pre intenzitný graf budeme mať počet stĺpcov zodpovedajúcich počtu agregovaných údajov v nich. Dáta sa budú zobrazovať v podobe grafu v užívateľskom rozhraní.

Formát údajov pre nastavenia a mobilitu bude dvojbodkou oddelený názov atribútu a jej hodnota.

Typy súborov budú:

- **Súbor s agregovanými údajmi** – obsahuje agregované údaje za čas definovaný užívateľom
- **Súbor s intenzitnými údajmi** – obsahuje viaceré agregované údaje
- **Súbor s nastaveniami merania** – obsahuje nastavenia aplikácie a mobilitu

### 3.2. Komunikačné protokoly

#### 3.2.1. Sériový port RS232

Komunikácia s mikrokontrolérom prebieha cez sériové rozhranie usb, tak, že jednotlivé bity prenášaných dát sú vysielané postupne za sebou (v sérii) po jednom páre vodičov v každom smere.

Arduino obsahuje sériovú triedu pre posielanie údajov na COM port pre hosťovský počítač. Štandardná sériová trieda riadená s Arduino IDE je SERIALASC pre ShieldBuddy.

Alokácia sériových kanálov je nasledovná:

SerialASC Arduino FDTI USB-COM micro USB

Serial1 RX1/TX1 Arduino J403 pins 17/16

Serial0 RX0/TX0 Arduino J403 pins 15/14

Serial RX/TX Arduino default J402 pins D0/D1

## 4. Návrh používateľského rozhrania

Dizajn návrhu zodpovedá ilustračnému obrázku definovanom v katalógu požiadaviek 3.1.1.

### 4.1. Názov projektu

A screenshot of a software interface showing a text input field labeled "Project name". The field is highlighted with a blue border and contains the text "Project name". Above the field is a small icon of a document with a red 'X' and the text "Project name".

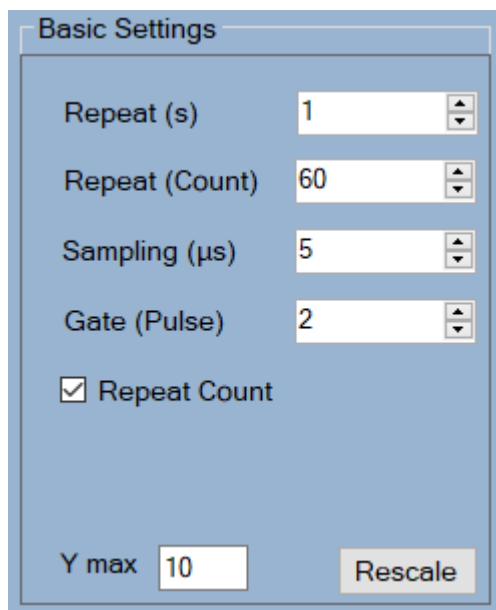
Meno projektu sa mení dynamicky, to, čo užívateľ napíše do textboxu bude meno projektu.

### 4.2. Nastavenia aplikácie

V danej sekcii užívateľ nastavuje parametre na základe ktorých bude ovplyvnený vypísaný graf, počet opakovaní v sekundách, počet meraní pre agregované dáta, impulz, ktorý spúšťa ióny do driftu v spektrometri a hustota merania.

Nastavujú sa parametre:

1. Repeat → zadávaný v sekundách (s)
2. Repeat → zadávaný počet opakovaní
3. Sampling → hustota merania, meraný v mý ( $\mu\text{s}$ )
4. Gate → veľkosť impulzu
5. Repeat Count → checkbox prepínajúci repeat z bodu 1 alebo 2

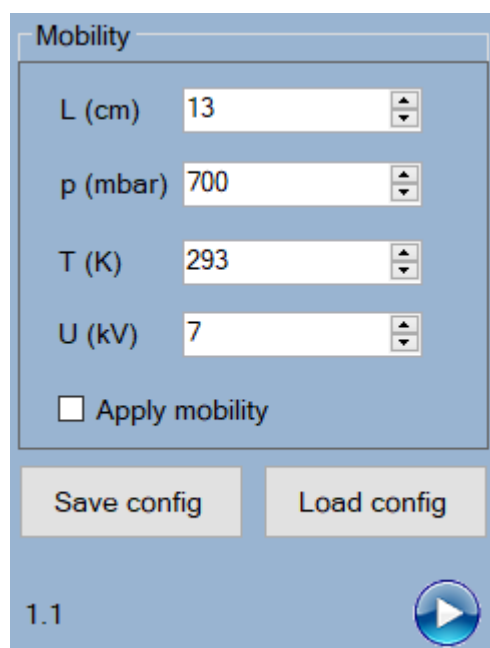
A screenshot of a "Basic Settings" dialog box. It contains several input fields with up/down arrows: "Repeat (s)" with value 1, "Repeat (Count)" with value 60, "Sampling ( $\mu\text{s}$ )" with value 5, and "Gate (Pulse)" with value 2. There is a checked checkbox labeled "Repeat Count". At the bottom, there is a "Y max" field with value 10 and a "Rescale" button.

Pri zvolení Repeat(Count) sa podľa zadáných parametrov vypočíta za aký čas prebehne simulácia. Táto informácia je len informačná pre užívateľa.

Grafu je možné aj zväčšiť či zmenšiť maximálnu hodnotu Y, hodnoty na osi X sú ovládané za pomoci myši.

#### 4.3. Nastavenie Mobility

Užívateľ nastavuje hodnoty pre arduino, presnejšie za akých podmienok sa budú vykonávať merania spektrometra.



Mobility

L (cm) 13

p (mbar) 700

T (K) 293

U (kV) 7

☐ Apply mobility

Save config Load config

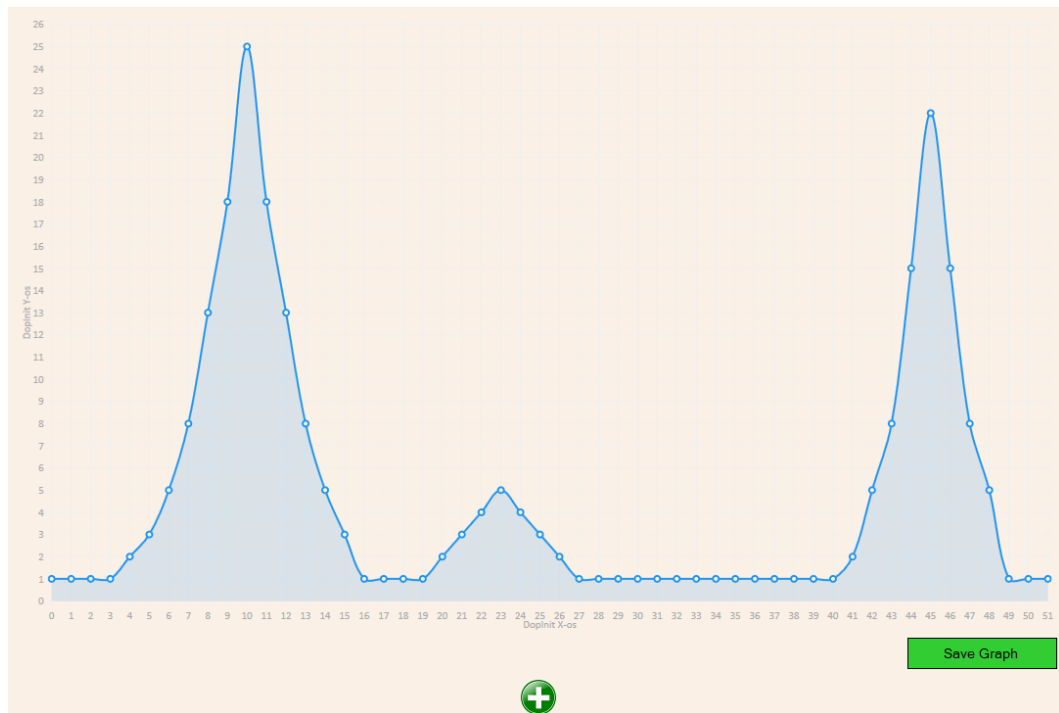
1.1

Nastavuje sa:

1. L → dĺžka meraná centimetromi (cm)
2. P → tlak meraný v mega baroch (mbar)
3. T → teplota v kelvinoch (K)
4. U → elektrické napätie merané v kilo voltoch (kV)
5. Apply mobility → aplikuje / neaplikuje mobilitu na dáta

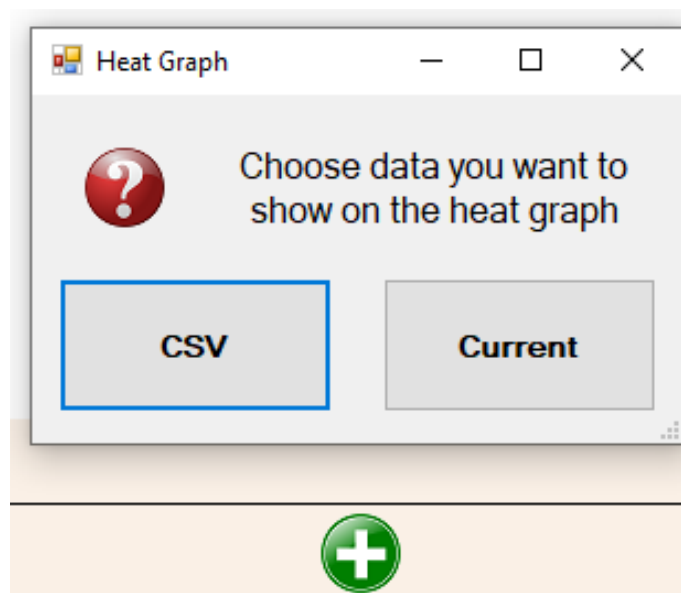
Okrem základných sekcií má užívateľ možnosť si uložiť konfiguráciu mobility pomocou tlačidla „Save config“, alternatívne možnosť načítať tieto údaje zo súboru tlačidlom „Load config“. Celé meranie sa spúšťa tlačidlom play, po ľavej strane od neho sa nachádza časovač, hodnoty sa budú rátať od 0 po nastavený Repeat zo sekcie Basic Settings.

#### 4.4. Sekcia grafu



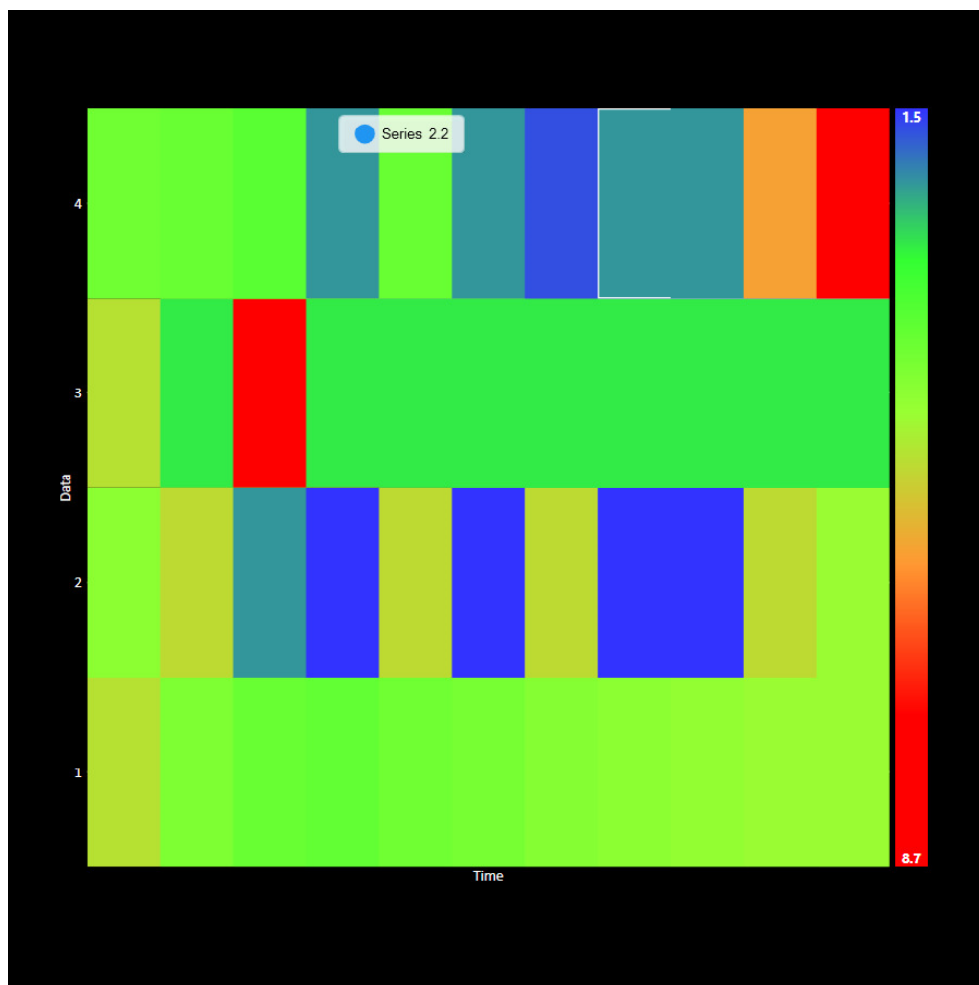
Na obrazovke je vykreslený graf z aktuálne nameraných dát. Jeho hodnoty a tvar sa menia v závislosti od parametrov spomenutých vyššie.

Tlačidlo plus ponúka možnosť vytvoriť intenzitný graf z aktuálne nameraných dát alebo už uložených do súboru \*.csv.



Po úspešnom načítaní dát a grafu sa intenzitný graf pripne pod hlavný graf, každý jeden heat graf je možno zatvoriť tlačidlom „Cancel” a v prípade, že bol vytvorený z aktuálne nameraných dát pribudne aj tlačidlo Save na uloženie agregovaných dát do csv súboru a tlačidlo „Stop/Start” podľa toho, či je vykresľovanie zapnuté alebo vypnuté. Ak užívateľ klikne

na ľubovoľný intenzitný graf vytvorí sa na pravej strane od jeho pozície graf obsahujúci údaje z daného riadku v zodpovedajúcom intenzitnom grafe.





## 5. Návrh implementácie

### 5.1. Class diagram

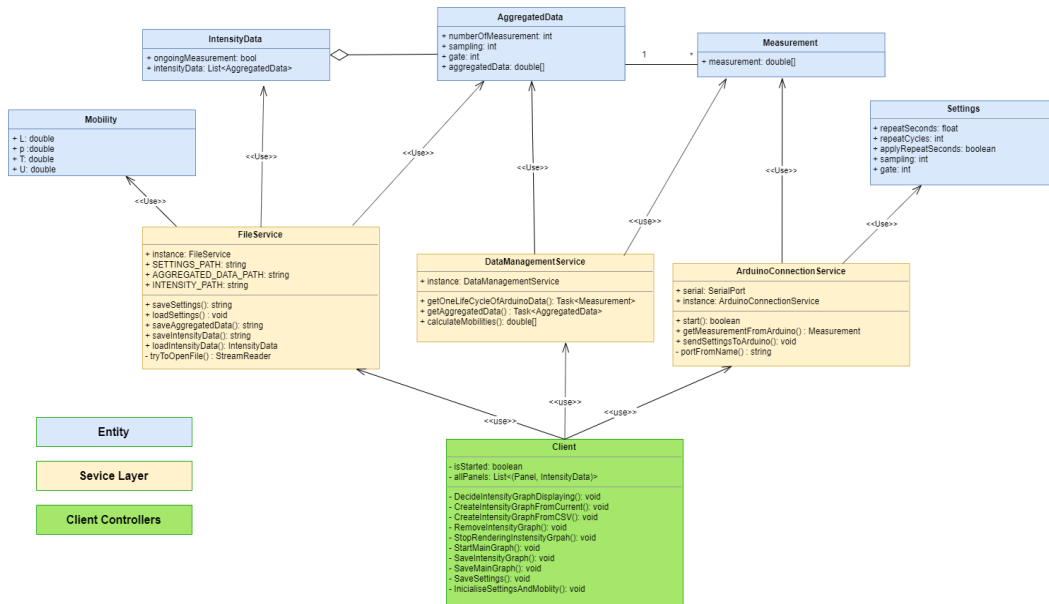
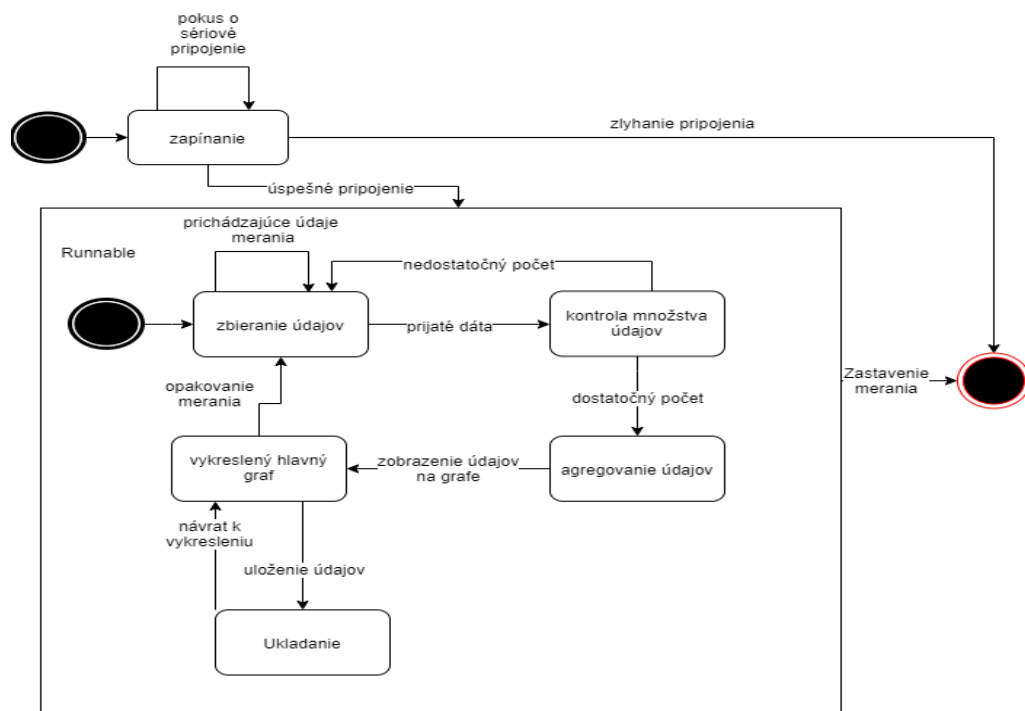


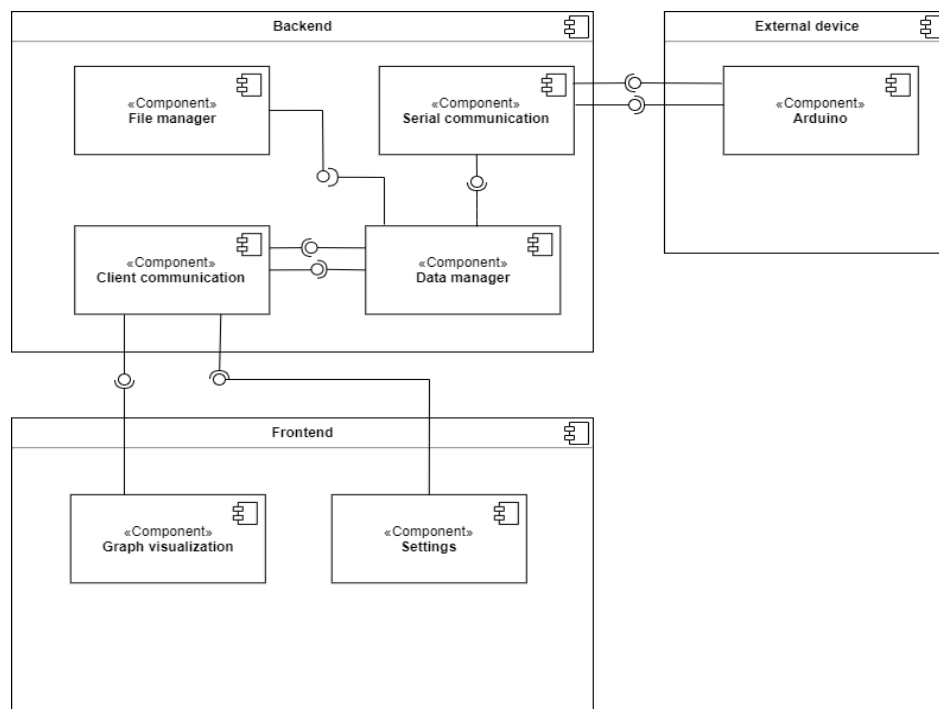
Diagram tried nasledujem architektúru zameranú na služby (service oriented architecture). Klient (zelená farba) je rozhranie komunikujúce s užívateľom, ktorý reaguje na diania (ako kliknutie na tlačidlo). Pre poskytnutie funkcionality nám pomáhajú služby (červená farba), ktoré fungujú ako singleton. Služby využívajú dátové štruktúry, entity ( modrá farba ) na uchovanie dočasných informácií.

### 5.2. Stavový diagram - Hlavný graf



Pri spustení aplikácie sa vytvorí pokus o sériové pripojenie k arduinu. Ak pripojenie zlyhá, aplikácie informuje chybovou hláškou užívateľa a končí. Ak sa pripojenie inicializuje, vytvorí sa vlákno, ktoré zbiera údaje z arduina. Hláškami Start a Stop arduino informuje začiatok a koniec merania. Pri správe stop sa skontroluje, či množstvo meraní presiahlo počet definovaným užívateľom. Ak áno, údaje sa agregujú a následne sa vykreslia v hlavnom grafe. Vlákno pokračuje v stave behu kým užívateľ nezastaví aplikáciu.

### 5.3. Diagram komponentov



Komponent diagram popisuje rozdelenie systému do samostatne fungujúcich častí. Základnou jednotkou nášho diagramu je backend a v ňom serial communication komponent, ktorý automaticky vyhľadá pripojené zariadenie (arduino), pokúsi s ním nadviazať spojenie a reaguje na príjem údajov. Prijaté údaje sa ďalej posielajú na data manager komponent, ktorý ich spracuje a využíva client communication komponent pre vykreslenie grafov v UI.