VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

KOMPIUTERIJOS KATEDRA

Kursinis darbas

**Pavadinimas**

Atliko: 3 kurso studentai:

Tomas Novikovas 4gr

Martynas Raila 1gr

Darbo vadovas:

Rytis Malakauskas

Vilnius

2016

# turinys

# Anotacija

LIETUVIŠKAI

The idea of this project is to create data acquisition and management system (DAMS) of EPC (Epilepsia Partialis Continua) patients. This system is going to be used by the doctors. The main requirements for DAMS was that the system has to be able to track all the medical history and treatment records about epilepsy patients. There also has to be invitation system, and three user types: administrator, coordinator, and doctor.

By using “Django” web framework and PostgreSQL database management system we have implemented login, registration, profile, and invite functionality. System allows to send three types of messages to users: private, public, and messages for coordinators. System has functionality that supports user management. Also the core functionality is added – case management. Users are able to add, edit, change approve setting on cases, export them to CSV files. System also allows to search for specific cases, view cases added by each user, and view statistics based on approved cases and follow-ups.

This report describes: database scheme, requirements, prototypes and implementation of “DAMS”.

# Sumarry

**<pavadinimas kursinio>**

ANGLIŠKAI

Mūsų projekto užduotis yra sukurti tarptautinę epilepsijos duomenų bazę („DAMS“ – duomenų kaupimo ir valdymo sistemą), kuria galėtų naudotis gydytojai. Sistema turi galėti sekti pacientų gydymo istoriją, vaistus, susitikimus su gydytojais, gebėti išsaugoti vaizdinę medžiagą apie pacientus. Sistema taip pat turi turėti tris vartotojų tipus: administratorius, koordinatorius ir daktaras. Buvo nuspręsta naudoti „Django” karkasą ir PostgreSQL duomenų bazių valdymo sistemą. Tinklapio išvaizda kurta pasinaudojus HTML ir CSS.

Yra sukurtas vartotojų valdymas. Įgyvendintos vartotojų registracijos, prisijungimų ir pakvietimų funkcijos. Sistema leidžia siųsti trijų tipų žinutes: privačias, viešas ir žinutes skirtas tik koordinatoriams. Taip pat pridėtas ir esminis funkcionalumas – ligos istorijų valdymas. Vartotojai turi galimybę pridėti, įkelti failą, keisti užpildytus duomenis ir pakeisti medicininės bylos patvirtinimo būseną. Sistema taip pat leidžia vykdyti bylų paiešką, ir peržiūrėti bylas, kurias pridėjo konkretus vartotojas. Taip pat galima peržiūrėti statistiką, kurioje duomenys atrinkti tik iš patvirtintų medicinos istorijų.

Šiame rašto darbe aprašyta duomenų bazės schema, reikalavimai, sistemos prototipai ir galutinis „DAMS” įgyvendinimas.

Mūsų projekto užduotis yra sukurti MIF superkompiuterio patalpos temperatūros bei drėgnio matavimo sistema bei ją aprašyti. Sistema turi susidaryti iš dviejų mažesnių sistemų. Pirmoji sistema bus sujungta su sensoriais ir matuos patalpos temperatūrą bei drėgnį. Antrąją sistemą sudaris serveris, kuriame „Cacti“ gaus duomenis iš pirmosios sistemos ir juos pateiks interneto naršyklėje diagramų forma. Mūsų sistema turi galėti matuoti temperatūra ir drėgnį, perduoti duomenis per vidinį tinklą pasinaudojus „SNMP protokolu bei panaudoti šiuos duomenis kuriant diagramas.

# Įvadas

LIETUVIŠKAI

Vision of the project is to create the data management and acquisition system (DAMS) of EPC (Epilepsia Partialis Continua) patients. System will store all needed information about patients including gender, diagnosis, family history, frequency of seizures, length of seizures, findings, successful treatments, unsuccessful treatments and more. Personal data like name or surname will not be stored. Also it was needed to create some kind of invitation system.

Nowadays doctors who deal with epilepsy disorder are using old and primitive way of collecting and sharing data about epilepsy patients. All the information about every patient is sent to one person using email and stored in Word or Excel files written in free-from. This method of storing information is not safe and extremely inconvenient. Our goal is to create a new data acquisition and management system (DAMS) for the doctors. Error: Reference source not found shows schema of DAMS. The idea was to create a website for DAMS. Once doctor was invited and registered, then doctors are able to log in into the website using correct credentials. Then they are redirected to the menu, where they can select to browse cases, add new one or to continue filling other case by adding follow-ups. Also there is an option to see all patients that doctor have already registered, and all doctors that agreed to join the system. All data that doctors filled will be saved into the database and will be accessible from anywhere through the website.

The purpose of this report is to represent and explain the idea, working principles, prototypes of the system, analysis and developing of the project. Section 1 describes requirements of the project. Section 2 describes design prototyping. In Section 3 development, and functions of the system are described. Section 4 describes testing. Also there are sections about conclusions and future works.

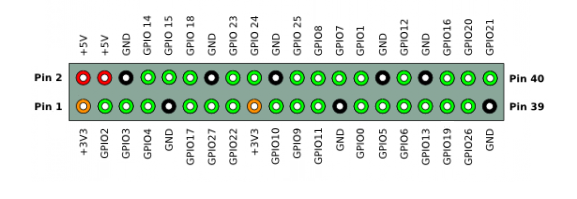
# Reikalavimai?

# Naudojama įranga (kažkaip gražiai pavadint)

reikia kažką parašyti

## Naudojama įranga

Termometrui naudojamas „Raspberry Pi 2 Model B“ kompiuteris. Jame įdiegta „Raspbian“ operacinė sistema. „Raspbian“ yra paremta „Debian“ operacine sistema optimizuota veikti su „Raspberry“ programine įranga.

„Raspberry Pi 2“ turi keturiasdešimt bendros paskirties įvesties ir išvesties jungčių, toliau GPIO, (GPIO – General Purpose Input Output), kurios naudojamos kaip sąsaja tarp kompiuterio ir kitų įrenginių. Iš keturiasdešimties jungčių (žr. 1 Pav.) dvi yra skirtos penkių voltų elektros srovės įtampai (raudona spalva), dvi – trijų voltų įtampai (oranžinė spalva), aštuonios jungtys – įžeminimui (juoda spalva) ir likusios 28 – įvesties ir išvesties duomenims siųsti (žalia spalva).

1 Pav. Bendros paskirties įvesties ir išvesties jungtys

Bendros paskirties įvesties ir išvesties jungtys yra numeruojamos dviem būdais: eilės numeriu ir jungties numeriu. Eilės numeriu yra numeruojamos visos jungtys, jungties numeriu tik jungtys skirtos siųsti arba gauti duomenis (žalia spalva). Visame apraše jungtys bus numeruojamos jungties numeriais, nes įtampos bei įžeminimo jungčių numeriai neturi reikšmės ir programos kode yra naudojami tik jungčių numeriai.

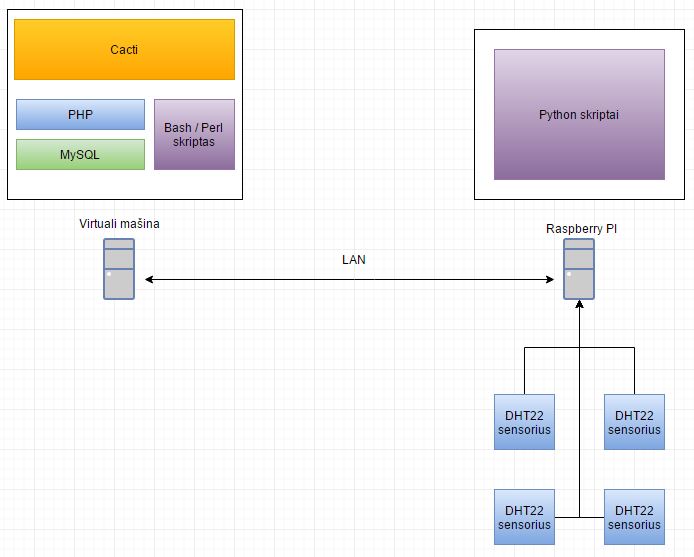
Temperatūrai ir drėgniui matuoti naudojami „AM2302/DHT22“ sensoriai. Sensoriai yra pritaikyti naudoti su 3 voltų ir 5 voltų elektros srovės įtampa. Maksimalus atstumas, kuriuo sensorius veikia yra šimtas metrų. Dažniausias intervalas, kuriuo gali būti renkami duomenys – dvi sekundes. Šie sensoriai pasirinkti dėl to, kad šių sensorių gavome iš projekto vadovo, jų funkcionalumo pilnai užtenka įgyvendinti projektą, yra pigūs bei turi biblioteką programavimui.

Sensorius turi keturias jungtis. Pirma skirta tiekti elektros srovę į sensorių, antra – duomenims gauti, o trečia ir ketvirta – įžeminimui. Nors sensorius turi dvi įžeminimo jungtis, užtenka naudoti tik vieną. Todėl projekto metu sensorių įžeminimui naudosime ketvirtą jungtį.

Sensorių jungimui prie kompiuterio naudojami vytos poros UTP kabeliai (angl. UTP twisted pair). Tokie kabeliai pasirinkti dėl to, kad juos gauname iš projekto vadovo ir jų pilnai užtenka įgyvendinti projektą.

## Naudojamos sistemos

Mūsų projektą sudaro dvi sistemos. Pirmoji – Raspberry PI naudojanti termometro sistema, kuri surenka duomenis iš prie jos prijungtų matavimo sensorių. Antroji – virtuali mašina, kurioje veikia „Cacti“ sistema, priimati duomenis iš pirmosios sistemos bei juos atvaizduojanti diagramose.

 Čia reikia nuorodos kaip virsui į paveikslėlį.

## Programavimo kalba ir bibliotekos

Termometrui kurti naudojama „Python“ programavimo kalba. Ji pasirinkta dėl to, kad oficialius sensnorių gamintojas „Adafruit“ pateikia biblioteką, parašyta šia kalba.

Bendrauti su sensoriais naudojama papildoma biblioteka „Adafruit\_Python\_DHT“. Tai biblioteka, sukurta specialiai DHT serijos sensoriams naudojamiems su „Raspberry Pi“. Biblioteka pritaikyta „Python 2.6/2.7“ versijai. Biblioteka yra nemokama. Licencija leidžia biblioteką bei su biblioteka susijusius failus naudoti, keisti, parduoti ar licencijuoti.

Pagrindiniai naudojami bibliotekos metodai yra read ir read\_retry. Tai metodai, kurie gauna sensoriaus tipą, jei jungties numerį ir grąžina iš sensoriaus nuskaitytą kortežą (angl. Tuple) su oro drėgniu ir temperatūra. Skirtumas tarp metodų tas, kad metodas „read“ iš sensoriaus duomenis skaito tik vieną kartą ir grąžina nuskaitytą kortežą, o metodas „read\_retry“ duomenis skaito penkiolika kartų ir grąžina pirmą nuskaitytą netuščią kortežą. Tokie du beveik vienodi metodai yra naudojami dėl to, kad pasitaiko atvejų, kai iš pirmo karto nepavyksta gauti sensorių nuskaitytos informacijos su temperatūra ir oro drėgniu ypatingai, kai sensorius yra sujungtas ilgesniais laidais. Informacijos iš sensoriaus pirmu kartu gauti nepavyksta dažniausiai dėl to, kad baigiasi atsakymo iš sensoriaus laukimo laikas.

# Termometro parengimas (pakeičiau, nežinau ar labai geriau)

Viena iš užduočių kursinio darbo metu yra ne tik suprogramuoti termometrą, kuris matuotų oro temperatūrą ir oro drėgnį VU MIF superkompiuterio patalpoje, bet ir sukonstruoti patį termometrą ir paruošti jį naudojimui. Termometras turi būti lengvai tobulinamas, t. y. turi būti nesunku pridėti papildomų dalių ar papildomų sensorių. Termometras vienu metu daugiausiai gali turėti keturis aktyvius sensorius dėl Raspberry PI jungčiu limitacijos. Papildžiau truputi

## Patalpa

Patalpa, kurioje stovi VU MIF superkompiuteris yra apytiksliai 8 metrų ilgio ir 5 metrų pločio. (PATIKSLINTI). Išilgai per vidurį patalpos stovi superkompiuteris ir visą patalpą padalina į dvi dalis, vadinamas karštąja ir šaltąja zonomis.

Karštoji zona yra ta pusė, į kurią išpučiamas šiltas oras iš superkompiuterio. Ši superkompiuterio pusė yra šiltesnė ir oro temperatūra yra šiek tiek aukštesnė, apie 20-24°C. Šaltoji zona yra ta pusė, kurioje superkompiuterio vėsinimui paduodamas šaltas oras ir kondicionierių. Temperatūra šioje zonoje siekia apie 14-19°C.

## Kompiuterio ir sensorių sujungimas

Sensoriai pastatyti keturiuose patalpos kampuose prie lubų. Du šaltojoje ir du karštojoje zonose. Kompiuteris, prie kurio laidais jungiasi sensoriai yra užkeltas virš pakabinamų lubų, netoli vieno sensoriaus karštojoje zonoje. Atstumai nuo kompiuterio iki sensorių yra apytiksliai: karštojoje zonoje – 1 metras ir 8 metrai, šaltojoje zonoje – 4 metrai ir 8 metrai. Žinoma laidai, skirti sujungti sensorius su kompiuteriu yra ilgesni, nes ne visi laidai nutiesti tiesiai iki sensoriaus, taip pat šiek tiek laido yra palikta atsargai.

Sensorių sujungti su kompiuteriu naudojame ilgus UTP kabelius ir trumpus pereinamus laidus su antgaliais, kurie tiek kompiuterio, tiek sensorių jungtis jungia su UTP laidu. Sensoriui sujungti reikalingi trys laidai. Čia reikia schemos

Sensoriui sujungti naudojami trys UTP kabelio laidai. Vienas elektros srovei, antras duomenims siųsti ir trečias – įžeminimui. Sutarta, kad kiekvienam elektros srovės laidui naudojame <> spalvos laidus, duomenims siųsti – <> spalvos laidus, o įžeminimui – <> spalvos laidus(čia reikia parašyti spalvas). Tokiu būdu nesunku susigaudyti, ką kuris laidas daro. Trys skirtingos spalvos naudojamos dėl to, kad ilgais atstumais siunčiant duomenis tuo pačiu laidu kaip ir elektros srovę, informacija gali būti pakitusi dėl susidariusių elektromagnetinių laukų.

## Prisijungimas prie kompiuterio

Prisijungti prie Raspberry Pi kompiuterio, stovinčio VU MIF superkompiuterio patalpoje, galima dviem pagrindiniais būdais. Pirmas būdas prisijungti prie kompiuterio yra naudojant monitorių ir klaviatūrą. Tačiau taip daryti nepatogu, nes kompiuteris guli gana aukštai. Antras būdas – prisijungti prie kompiuterio nuotoliniu būdu naudojant SSH (Secure Shell) protokolą. Raspberry Pi kompiuteris savo programinėje įrangoje turi pagal nutylėjimą įjungtą SSH serverį. Prisijungti prie kompiuterio galima tik naudojant kitą kompiuterį, esantį tame pačiame tinkle. Tokiu būdu prisijungus prie Raspberry Pi kompiuterio galima naudotis tik komandine eilute, tačiau to pilnai užtenka.

# Termometro programavimas

## Sensorių informacijos skaitymas

kaip skaito info iš sensorių

## Informacijos saugojimas

loginimas į failą

## Pranešimų siuntimai

email ir sms žinutės

## Nustatymų failas

settings ir parametrai

# Duomenų perdavimas į virtualią mašiną

Sistemoje buvo įrašyti „snmp“ ir „snmpd“ programų paketai. „SNMP“ konfigūracijos faile buvo pridėta papildomi parametrai, kuriuose buvo nurodyta IP adresas, į kurį bus siunčiami duomenys (Virtualios mašinos IP adresas), bei komanda su unikaliu „OID“, kuri paleis „Python“ skriptą sistemoje, gražinantį duomenis iš sensorių. „OID“ yra unikalus objektą identifikuojantis numeris, naudojamas atskirti „snmp“ žinutes. Pasinaudojus „snmpwalk“ funkcija bei komandai priskirtu „OID“ terminale atrenkame vieną tikslų „OID“, kuris gražiną mums reikalingus duomenis. Šis „OID“ bus naudojamas gauti duomenims iš „Raspberry PI“ virtualios mašinos „Bash / Perl“ skripte.

# Virtualios mašinos konfigūracija

# Pagrindinių programinų paketų instaliacija

Virtualioje mašinoje pirmiausia buvo įrašytas „cacti-spine“ programų paketas. Jis įrašo visus papildomus programų paketus reikalingus funkcionaliam „Cacti“ darbui („apache2“, „PHP 7“, „MySQL 5.2“). Instaliacija turi patogų įrašymo procesą ir yra intuityvi, leidžianti sukurti visus reikalingus vartotojo prisijungimus prie programų instaliacijos procese. Taip pat buvo įrašyti „snmp“ ir „snmpd“ programų paketai, leidžiantys gauti bei siūsti informacija „SNMP“ („Simple Network Management Protocol“) protokolu. „SNMP“ konfigūracijos faile buvo pridėta papildomi parametrai, kuriuose buvo nurodyta IP adresas, iš kurio bus gaunami duomenys (“Raspberry PI” IP adresas).

# Duomenų gavimas iš Raspberry PI

Virtualioje mašinoje taip pat buvo sukurtas „Bash / Perl“ skriptas, kuris kreipiasi į „Raspberry PI“ per „SNMP“ protokolą. Pasinaudojus „snmpget“ funkcija bei unikaliu „OID“, gaunami duomenys iš „DHT22“ sensorių. Duomenys po to yra „apdirbami“ Perl kodo pagalba, kad jie turėtų tokį formatą: „pavadinimas:vartė pavadinimas:vertė pavadinimas:vertė ir t.t.“. Šioje vietoje Perl kodas pasirūpina, kad duomenų formatas būtų priimtinas „Cacti“ sistemai.

## „Cacti“ konfigūracija

Naršyklėje įvedus “http://(virtualios mašinos IP adresas)/cacti”, vartotojas pasiekia internetinę “Cacti” instaliaciją, kurioje sistema parodo reikalingus ir įrašytus programų paketus, bei leidžia pasirinkti, kokias “Cacti” komponentų versijas norėsime naudoti. Pabaigus instaliaciją, prie “Cacti” sistemos galima prisijungti suvedus atitinkamus prisijungimo duomenis. Pirmą karta prisijungiant reikia įvesti iš anksto numatytus vartotojo duomenis (vartotojas: admin, slaptažodis: admin). Po prisijungimo vartotojas yra priverstas pasikeisti savo slaptažodį.

# Diagramų kūrimas „Cacti“ sistemoje

Cia turi buti 6

Pirmiausia sukuriame naują datos įvedimo metodą (Data Input Methods). Įrašome pavadinimą (Title), pasirenkame įvesties tipą (Input type) „Script/Command“, ir nurodome kelią failų sistemoje iki skripto, kuris mums gražiną sensorių duomenis (Input String). Paduodant papildomus parametrus mūsų skriptui, reikia prie kelio iki failo pridėti papildomus kitamuosius. Tai galėtų atrodyti taip: „/path/to/script/data.sh <ip>“. Nurodžius papildomus parametrus reikia kiekvienam iš jų pridėti papildoma įvedimo laukelį (Input Field). Jam suteikti pavadinimą ir „draugišką“ pavadinimą. Kiekvienai mūsų gaunamai duomenų “pavadinimas:vertė” porai sukuriame po išvesties laukelį (Output Field) su pavadinimu ir „draugišku“ pavadinimu.

Toliaus sukuriam naują prietaisą (Device). Įvedame prietaiso aprašymą/pavadinimą (Description), IP adresą (Hostname). Pasirenkame šeimininko šabloną (Host Template), šiuo atvėjų „Local Linux Machine“. Papildomų parametrų keitimas priklauso nuo situacijos ir nėra būtinas.

Sekantis žingsnis sukurti datos šaltinį (Data Source). Pasirenkame šeimininką (Host) iš pateikiamo sąrašo. Suteikiame šaltiniui pavadinimą (Name). Pasirenkame datos įvedimos metoda (Data Input Method) iš esamo sąrašo. Kiekvienam esamam įvedimo laukeliui (Input Field) datos įvedimo metode priskiriame vertę. Kiekvienam išvedimo laukeliui (Output Field) sukuriam naują duomenų šaltinio elementą (Data Source Item), kuriame įvedame vidinį duomenų šaltinio pavadinimą (Internal Data Source Name) ir prie išvesties laukelio (Output Field) pasirenkame norimą duomenų porą, kuri priklausis šiam duomenų šaltinio elementui (Data Source Item).

Atlikus šiuos žingsnius galima be vargo sukurti duomenis atvaizduojančias diagramas. Pasirinkę diagramų valdymą (Graph Management), galime sukurti naują diagramą arba keisti jau esamas. Išsirenkame iš sąrašo šeimininką (Host). Įvedame pavadinimą (Title). Reikalui esant galima keisti papildomus parametrus pagal poreikius. Norint pavaizduoti duomenis diagramoje, turime pridėti naują diagramos elementą (Graph Item). Jam pasirenkam šeimininką (Host) iš sąrašo. Iš duomenų šaltinių (Data Source) sąrašo pasirenkame mus dominančius duomenis, kuriuos apsirašėme išvesties laukeliuose (Output Field) datos šaltinyje (Data Source). Pasirenkame duomenis atvaziduojančią spalvą (Color), diagramos elemento tipą (Graph Item Type) ir konsolidacijos funkciją (Consolidation Function).

# Testavimas

Sistemos testavimas buvo atliktas virtualioje aplinkoje. Tam buvo pasitelkta virtualizacijos programa „VirtualBox“, kurios pagalba buvo sukurtos dvi virtualios mašinos su „Ubuntu 16.04“ operacine sistema. Abi mašinos naudojo „bridged network“ tinklo adapterį, tam, kad gautu invidualius vidinius IP adresus bei galėtų pasiekti internetą. Pirmoji mašina įmitavo „Cacti“ serverį, o antroji – Raspberry PI kompiuterį.

Pirmieji testai nedavė jokių rezultatų. Po ilgų ieškojimų buvo atrasta, kad duomenys, kurie pasieki a vidinę „Cacti“ sistemą turi turėti griežtą formatą. Kadangi duomenys yra „String“ formato, jie turi būti pateikiami tokia forma: „pavadinimas:vertė pavadinimas:vertė pavadinimas:vertė ir t.t.“. Tačiau šių duomenų pabaigoje negali būti nei tarpo ženklo, nei naujos eilutės ženklo.

Taip pat nemažai laiko buvo skirta testuoti duomenų perdavimui per „SNMP“ protokolą. Kadangi nei vienas iš mūsų nebuvo susiduręs su šiuo protokolu tiesiogiai, užtrukome, kol teisingai sukonfigūravome abi mašinas, kad „Cacti“ serveris galėtų kreiptis į Raspberry PI, o šis jam gražintų duomenis. Šioje vietoje daug padėjo „log“ failų stebėjimas ir pateiktos informacijos analizavimas.

# Išvados