Python-Cyberus

Raziskovalna naloga

Mentor:
Boštjan Lubej

Avtor:
Jože Lavrič

Celje, maj 2021

IZJAVA

IZJAVA

Mentor Lubej Boštjan v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Python-Cyberus, katere avtor je Jože Lavrič

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 13.5.2021

ŠOLSKI CENTER CELJE
SREDNJA ŠOLA ZA KEMIJO, ELEKTROTEHNIKO IN RAČUNALNIŠTVO
CELJE, Pot na Lavo 22

Podpis mentorja

Podpis odgovorne osebe

i

ZAHVALA

Za nastanek raziskovalne naloge in njeno uspešno izvedbo se zahvaljujem vsem, ki so mi na moji poti pomagali, mi podali strokovne kritike in nasvete ter me spodbujali.

Posebno zahvalo bi rad namenil mentorju prof. Boštjanu Lubeju, ki me je pri izdelavi naloge vodil in usmerjal, da sem pri razvijanju aplikacije ter pisanju raziskovalne naloge izkoristil svoj celoten potencial.

Seveda pa ne smem pozabiti še na vse tiste, ki so si vzeli čas, da so rešili anketo, zagnali moj program ali prebrali raziskovalno nalogo in na vse ostale profesorje, ki so mi v zadnjih letih pomagali priti do novih znanj in rasti.

POVZETEK

KLJUČNE BESEDE: Python, podatki, Urwid

V svoji nalogi obravnavam vidik komunikacije med različnim sistemi. Osredotočil sem se predvsem na posebno vrsto komunikacije: med odjemalcem in strežnikom.

Naloga odgovarja na teoretična vprašanja, kot so, kaj je dokumentacija podatkov in kako te ohranimo varne, kaj je komunikacija med sistemi ter zakaj je oboje pomembno. Teoretično povzamem jezik Python, jezik, s katerim sem implementiral svojo rešitev. Veliko vlogo pri grajenju programa je igrala tudi knjižnica Urwid, kateri prav tako namenim nekaj besed.

V praktičnem delu predstavim, kako sem komunikacijo med sistemi implementiral v svoji rešitvi. Sprogramiral sem terminalsko aplikacijo, ki predstavlja odjemalca ter strežnik, ki z njim komunicira. Komunikacijo med njima predstavlja želja po shrambi in dokumentaciji strojnih in mrežnih podatkov računalnika.

Zaradi radovednosti sem raziskal, kako koristen bi bil tak program drugim. Pri tem se osredotočil tako na podjetja, kot na posameznike, ki imajo računalnike tudi za osebno rabo. Zanimalo me je, če jim je dokumentacija pomembna oziroma če podatke sploh na kakšen način hranijo in če bi jim moj program prinesel dodatno vrednost. Rezultate sem analiziral.

SUMMARY

KEY WORDS: Python, data, Urwid

In my thesis, I address the aspect of communication between different systems. I focused mainly on a specific type of communication: between the client and the server.

The project answers theoretical questions such as what data documentation is and how to keep it secure, what communication between systems is, and why both are important. Theoretically, I summarize the Python language, the language with which I implemented my solution. The Urwid library also played a big role in building the program and that is why I also dedicate a few words to it.

In the practical part, I present how I implemented communication between systems in my solution. I programmed a terminal application that represents the client and a server that communicates with it. The communication between them is represented by the desire to store and document computer hardware data and network data.

Out of curiosity, I researched how useful such a program would be to others. In doing so, I focused on both organizations or companies and individuals who also have computers for personal use. I was interested if documentation was important to them or if they store various data in any way at all and if my program would bring them any additional value. I analyzed the results.

Kazalo vsebine

Kazalo slik

1 UVOD

Komunikacija je v življenju zelo pomembna, lahko bi rekli, da je najpomembnejša za delovanje družbe. Komuniciramo z različnimi ljudmi v različnih oblikah: elektronsko, pisno, ustno, preko gest, tona našega glasu in še na toliko več drugih načinov. Čeprav je komunikacija z ljudmi izjemno pomembna, je tako pomembna tudi komunikacija v sistemih.

Komunikacijski strežnik je namenski sistem, ki zagotavlja komunikacijske storitve za uporabnike v omrežju, ki morajo prek telekomunikacijskih povezav prenašati datoteke ali dostopati do informacij v sistemih ali omrežjih na oddaljenih lokacijah.

Komunikacija s serverji se deli na tri dele:

- Zahteva (Requests)
- Odziv (Response)
- Storitev (Service)

Zahteva se pošlje od odjemalca, da od strežnika zahteva nekatere podatke, kot so datoteke, ali pa mu sporoči kaj se naj zgodi.

Odziv pošlje odgovor iz strežnika k odjemalcu glede na zahtevo odjemalca.

Storitev je posebna naloga, ki jo strežnik zagotovi za uporabo odjemalca.

Komunikacijo s strežnikom lahko tako nekako predstavimo z obiskom zdravnika. Pri tem bi bolnik bil odjemalec in zdravnik strežnik. Odjemalec bi zaprosil zdravnika, da mu pove, kaj je z njim narobe in mu poleg podal informacije v obliki simptomov. Zdravnik bi zahtevo prejel in mu vrnil odgovor v obliki diagnoze in receptov za tablete. Storitev, ki nam jo zdravnik pri tem zagotavlja, je zdravniški pregled.

Komunikacija je prav tako pomembna, ker omogoča drugo pomembno zadevo, ki se je želim dotakniti v projektni nalogi – dokumentacijo.

Dokumentacija je beleženje dogodkov, podatkov in informacij, ki so pomembne za organizacijo, zgodovino in iskanje rešitev. Tako lahko beležimo napake, ki nam bodo pomagale odkriti težavo v programu. Beležimo lahko datume, ki označujejo pomembne dogodke in zapišemo lahko zgodovino vseh imen uporabnika, ki nam bodo pomagale identificirati njegovo preteklo aktivnost.

Kako lahko nekaj shranimo, če sistem drugemu sistemu ne zna povedati kaj naj shrani, v kakšni obliki in kje?

Zato sem se odločil združiti ta dva pomembna vidika in razviti aplikacijo, ki vse računalniške podatke pošlje na poseben strežnik, kjer so vsi podatki varno shranjeni.

1.1 Hipoteze raziskovalne naloge

Hipotezo sem razdelil na dva dela in sicer, prvi del je kaj si mislijo in ali bi uporabljali povprečni ljudi. Drugi del je, če bi moj program uporabljala podjetja, ki se z računalništvom ukvarjajo.

Hipoteze za povprečne uporabnike so:

- Večino povprečnih uporabnikov to ne bi zanimalo.
- Večina povprečnih uporabnikov nima več računalnikov.

Hipoteze za podjetja so:

- Večina podjetij bi zanimalo moj program.
- Večina podjetij ima več kot deset računalnikov.

1.2 Metode dela

Pri raziskovanju sem uporabil več metod dela in sicer:

- Metodo ankete, kjer sem sestavil dve ankete, eno namenjena splošnim uporabnikov računalnikov in ena namenjena podjetjem.
- Metodo raziskovanja, kjer sem obdeloval različne literature glede programskega jezika Python ter komuniciranja.

1.3 Moji cilji

Cilj moje raziskovalne naloge je bil narediti program, ki bi olajšal shranjevanje podatkov različnih računalnikov na skupnem strežniku.

2 TEORETIČNI DEL – PYTHON

Za Python sem se odločil, ker se mi je zdel najboljši jezik, za to, kar potrebujem. Potreboval sem jezik, ki lahko ustvarja TUI (Text User Interface). Raziskal sem tudi, kaj so uporabili ustvarjalci Linux Ubuntu-a, in sem ugotovil, da so uporabili Python.

Prednost Python-a je to, da ima veliko podporo in skupnost. Ima tudi veliko knjižnic, ki imajo podporo in jo še bodo imele v prihodnosti.

2.1 Kaj je Python?

Python je visoko ravni večnamenski programski jezik. Ustanovil ga je matematik Guido van Rossum leta 1990. Podpira dinamične podatkovne tipe, zato je drugačen od Jave ali C programskih jezikov in je bolj podoben programskemu jeziku kot je Ruby.

Uporablja strukturiran in objektno orientiran programski stil.

2.2 Python knjižnice

Python ima ob namestitvi osnovne knjižnice. Knjižnice so skupek programske kode, po navadi so to različne funkcije, katere lahko programer uporabi v svojem programu. Nekaj teh knjižnic je napisanih v jeziku C.

Standardne knjižnice:

- Math,
- Random,
- Datetime,
- Sys,
- Os.

Trenutno obstaja več kot 270.000 knjižnic za Python. Vsaka Python verzija ne podpira vseh knjižnic in obratno. Če je knjižnica napisana za Python 3, je zelo mala verjetnost, da bo delala tudi na Python 2. Knjižnice, ki niso del standarda se imenujejo knjižnice tretjega nabora, ene izmed teh so:

- Pygame,
- Matplotlib,
- Numpy,
- Pandas.

Razlika med standardnimi knjižnicami in knjižnicami tretjega nabora je, da so naložene preko portala PyPi (Python Package Index).

Python Package Index je skladišče programske opreme oziroma knjižnic za Python. PyPi pomaga najti knjižnice, s PIP komando, s katero upravljamo želene pakete v projektu, pa jih inštaliramo direktno iz skladišča PyPi.

2.3 Uporaba Python-a

Python je uporabljen za različne tipe aplikacij, zato ima tudi različna orodja. Uporablja se za različne aplikacije: za razvijanje internetnih aplikacij, tako sočelja kot strežniških storitev, za

upravljanje strojne opreme, belo oziroma etično »hekanje«, zelo popularno pa je tudi analitično delo in strojno učenje ter umetno inteligenco.

Python je zelo popularen in ga uporabljajo tudi NASA, Pixar, Google, Facebook, YouTube in še mnoge druge organizacije.

3 TEORETIČNI DEL – Urwid

Urwid je knjižnica za Python, ki omogoča izdelavo TUI.

TUI oziroma Text-based User Interfaces se uporablja za prikazovanje in ne potrebuje posebnih grafičnih vmesnikov. Tako lahko deluje normalno v terminalu. Zato sem se za to knjižnico tudi odločil, ker lahko z njo izdelamo program, ki deluje na katerem koli računalniku, ki ima terminal. Prav tako za zagon takšne aplikacije ne potrebujemo velike procesne moči. Curses je še ena takšna knjižnica, ki deluje podobno kot Urwid. Na koncu sem se odločil zanj, saj lahko z njim narediš več in je bolj kompleksen.

Urwid ima že vgrajene module, ki jih lahko spreminjaš in urejaš. Podpira 24-bitno in 256 barvno nastavitvijo za ospredja in ozadja, seveda pa je to odvisno tudi od terminala v katerem program zaganjaš.

Ko modul upodobi platno, ki ga more narisati na zaslonu, se v predpomnilniku shrani referenca nanj. Ta predpomnilnik pa se uporabi vsakič ko je potrebno, tako se zmanjša količina dela, ki je potrebna za posodobitev zaslona.

Urwid ima glavno zanko, ki pa se deli na:

- Prikazovalni modul, katerega naloga je nadzirati ekran, tipkovnico in miško.
- Postavitev modulov, ki skrbi za postavitev besedila in vsebino listov.
- Zanke dogodkov, ki pa skrbijo za merilnike časa in način izvajanja dogodkov.

4 PRAKTIČNI DEL

4.1 Anketiranje

Pri anketiranju sem ciljal na dve skupine in sicer: ljudi, ki so splošni uporabniki računalnikov in podjetja, katera imajo več računalnikov in se ukvarjajo s tehnologijo.

4.2 Glavni program – Cyberus

Opazil sem, da na trgu ni veliko programov, ki ti pomagajo organizirati podatke svojih računalnikov. Zato sem se odločil, da bom takšen program naredil sam.

Program je narejen z namenom, da olajšamo in izboljšamo delo IT osebja. Delavec prižge program, izpolni potrebne podatke in jih pošlje. Nato dobi strežnik vse te podatke in jih pravilno shrani.

4.3 Izdelava programa

Program sem naredil v Python 3.6, uporabil pa sem tudi naslednje knjižnice:

- Binascii,
- Glob,
- Urwid,
- Json,
- Keyboard,
- Logging,
- Subprocess,
- Base64,
- Os,
- Encodings,
- Socket,
- Time.

Binascii je knjižnica, ki sem jo uporabil za možnost pisanja v NVRAM, ki predstavlja del pomnilnika. Ta možnost v mojem programu ni aktivirana se pa lahko aktivira če so takšne potrebe.

Glob knjižnica se uporablja za iskanje datotek v vašem računalniku. To sem potreboval, da sem lahko odprl datoteko formata json in jo prebral.

Urwid sem že namenil nekaj besed. Je glavna knjižnica, ki sem jo uporabil. Potrebna je za vizualni del mojega programa.

Keyboard je knjižnica s katero je mogoče poslušati kaj uporabnik pritisne na tipkovnici. Zaradi te knjižnice lahko uporabnik pritisne tipko »esc« in s tem zapre program. Slaba lastnost te knjižnice je, da more program zagnati skrbnik, da knjižnica pravilno deluje.

Logging knjižnica se uporablja za beleženje dogodkov v programu. Z njo sem ustvaril datoteko, v kateri se dokumentirajo vse stvari, ki se naredijo v programu. Uporabil sem jo tudi, da sem to ustvarjeno datoteko prebral in njeno vsebino zapisal v posebnem »log« okenčku.

S knjižnico **Subprocess** lahko zaženemo komande v terminali. Tako pridobimo nekatere podatke računalnika, kot so na primer verzija bios-a, različni IP-ji in dostopni porti.

Base64 je knjižnica s katero lahko zakodiramo besedilo v base64 format. To ni tako dobro za varnost podatkov, kakor je za hitrejše in bolj varno (saj je manj možnosti da pride do napake) pošiljanje besedila.

Os sem uporabil, za preverjanje ali specifična datoteka obstaja na računalniku.

Za pravilno formatiranje podatkov, ki se zapišejo v NVRAM sem uporabil **Encodings.** Ta podpira UTF-8, kar je potrebno, če želimo datoteko formata text pretvoriti v HEX in jo zapisati v NVRAM.

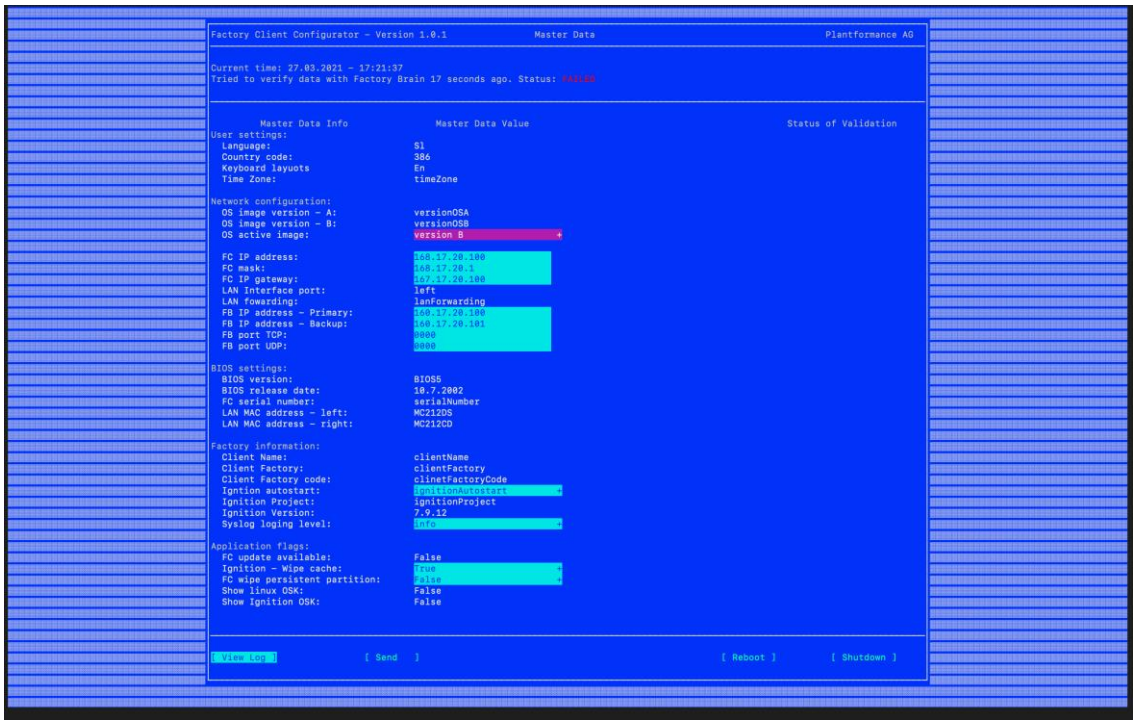
Socket sem uporabil za pošiljanje datoteke json formata na strežnik in v strežniku samem, za sprejemanje te datoteke oziramo besedila.

Time sem uporabil, da lahko dobim čas sistema in ga izpišem ter da sem lahko naredil samodejno pošiljanje podatkov vsakih 30 sekund.

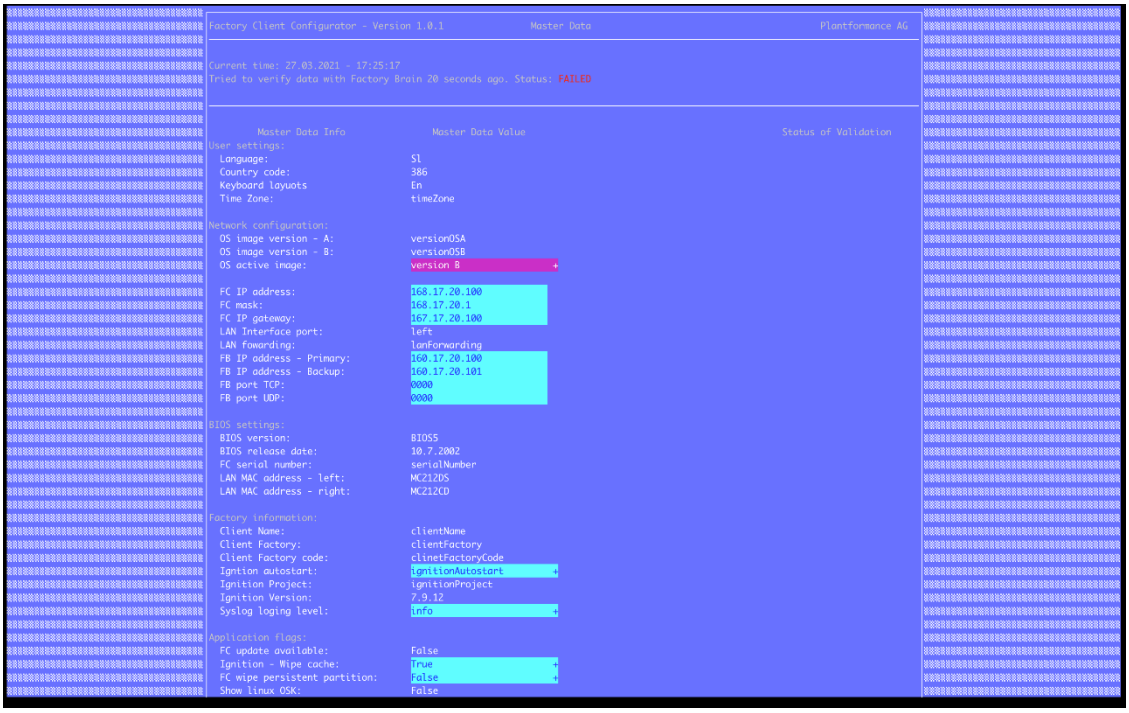
4.4 Izgled programa

Program se zažene v terminalu, zato temu primerno tudi izgleda. Kako program izgleda, pa ni samo odvisno kako je program napisan ampak tudi v kakšnem terminalu je zagnan. Nekateri terminali ne podpirajo barv ali pa podpirajo le nekatere. Najpogosteje podpirajo 8, 16 ali 256 barv. Razlike pa se ne končajo pri barvah, tudi koliko programa se lahko pokaže na ekranu je odvisno od terminala do terminala.

Te razlike lahko vidimo pri terminalu od Mac OS-a (Slika 1 Mac OS – Terminala) in iTerm-a (Slika 2 iTerm).



Slika 1 Mac OS – Terminala

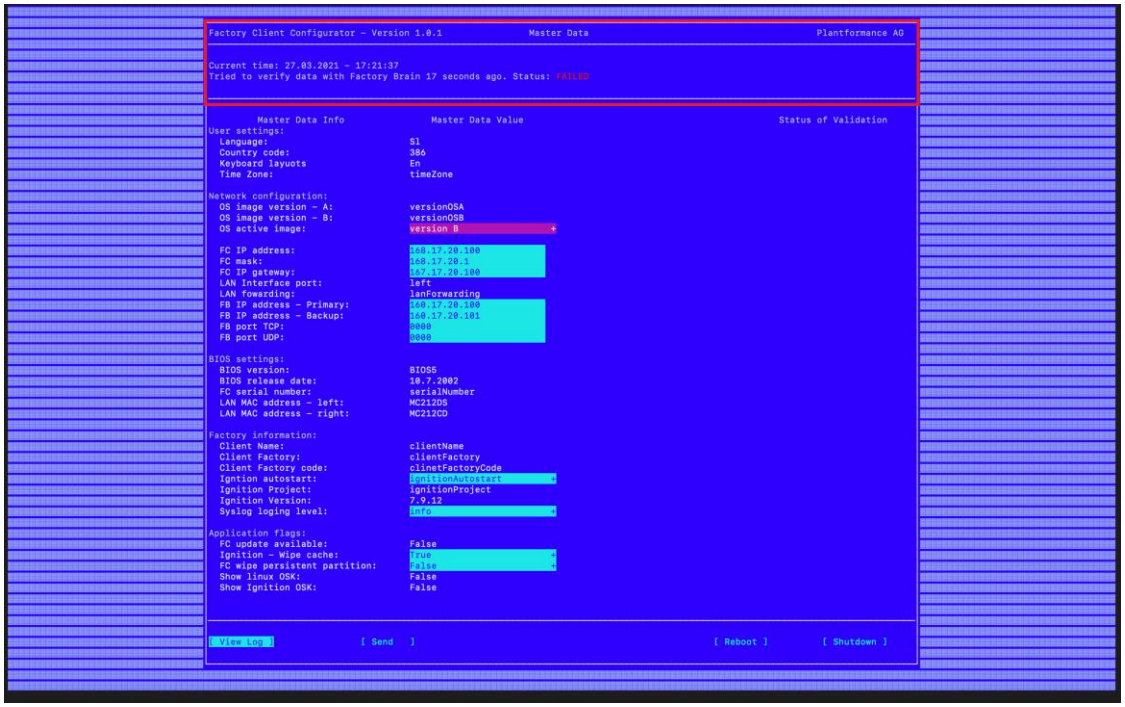


Slika 2 iTerm

4.5 Sestava programa

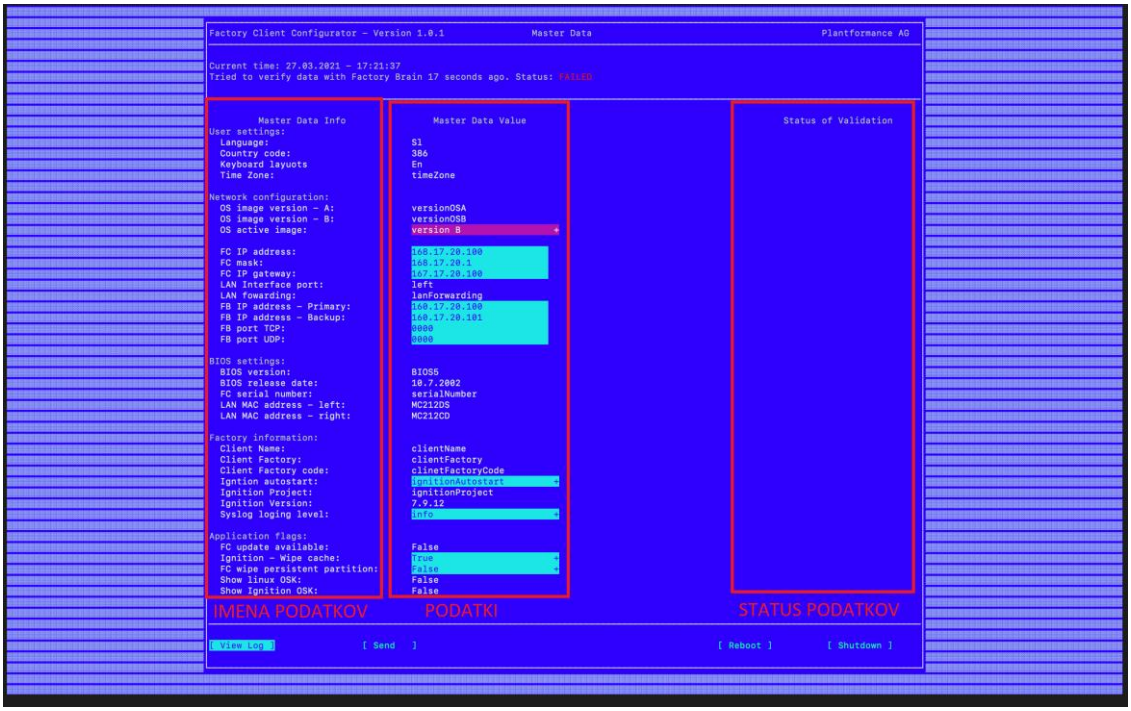
Program je sestavljen iz različnih oken. Začetno okno je že bilo predstavljeno, imamo pa še okno za log in manjša okna za izbiranje ter vpisovanje podatkov.

Začetno okno je razdeljeno na glavo (Slika 3 Glava programa), kjer je naslov okna, podatki o programu (verzija programa), čas in status pošiljanja.



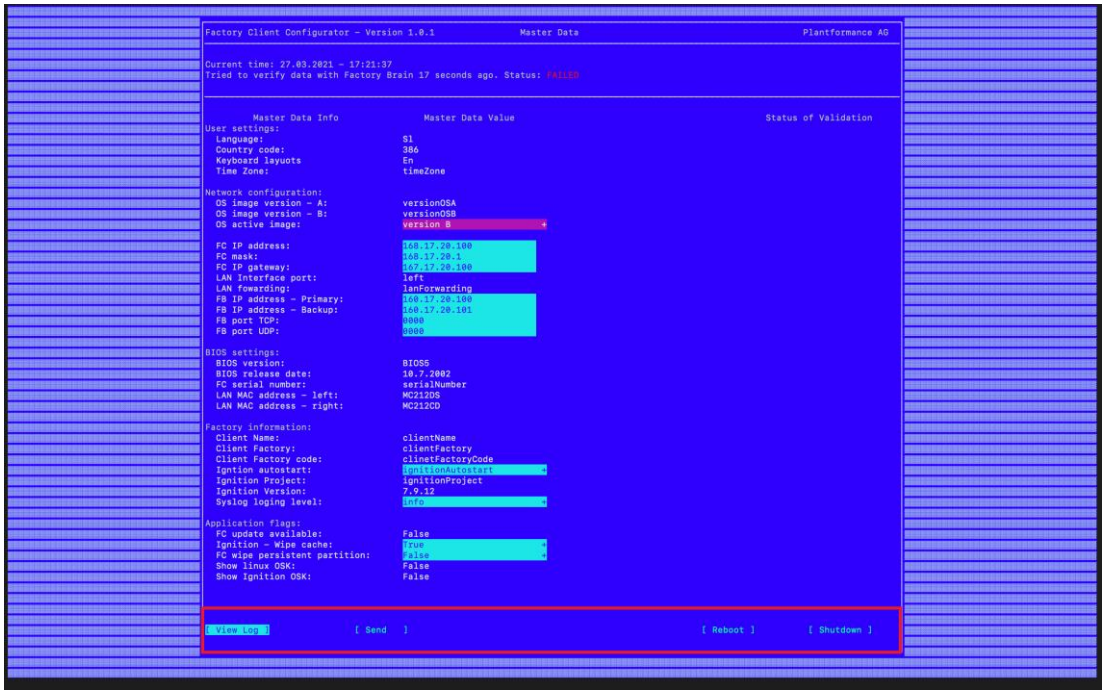
Slika 3 Glava programa

Pod glavo je telo, katerega sem razdelil na tri dele: imena podatkov, podatki in status podatkov. Izgled telesa z vsemi tremi sestavnimi deli prikazujem v spodnji sliki (**Napaka! V ira sklicevanja ni bilo mogoče najti.**).



Slika 4 Telo programa

Nazadnje moj program sestavlja še noga (Slika 5 Noga programa), v kateri najdemo tipke »View log«, »Send«, »Reboot« in »Shutdown«.

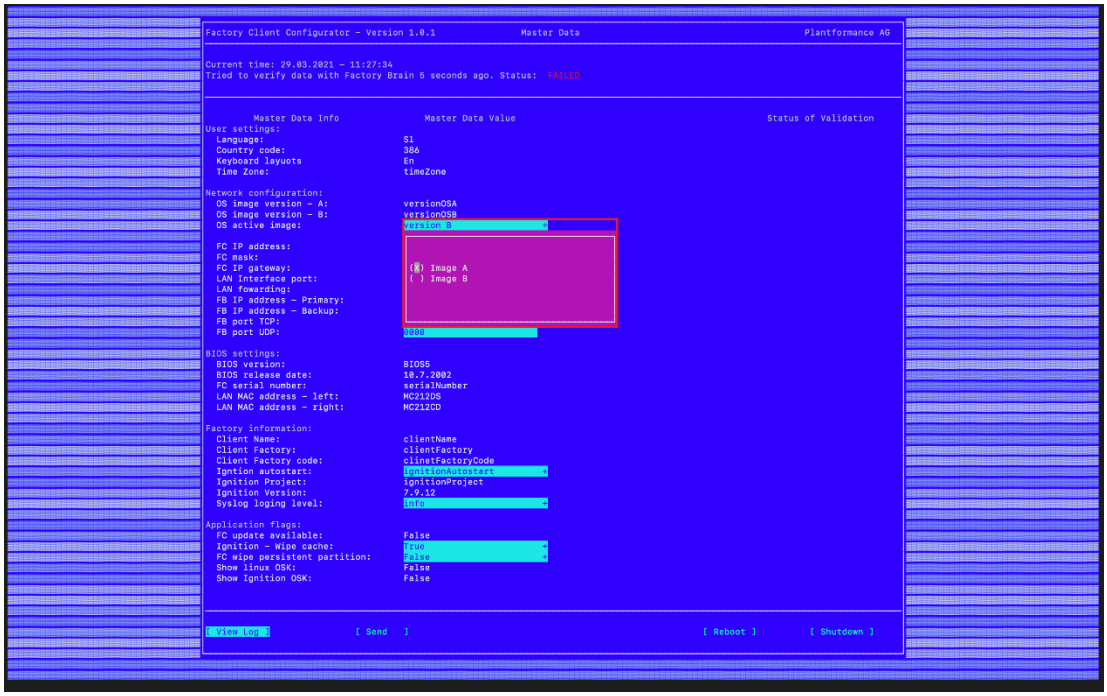


Slika 5 Noga programa

4.6 Delovanje programa

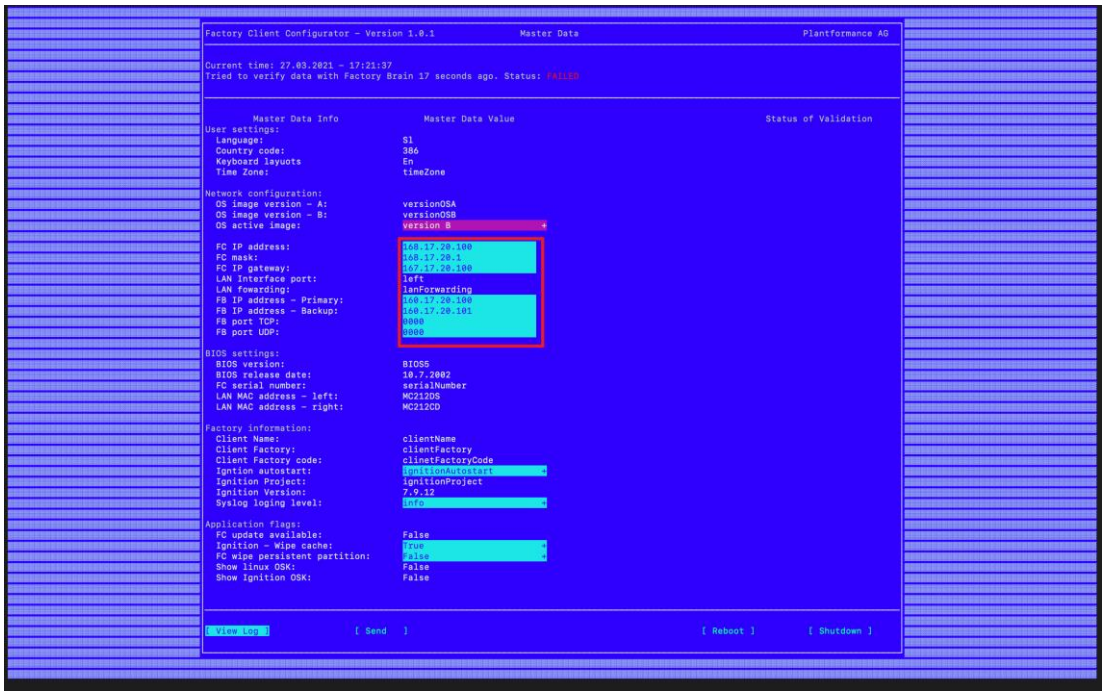
Kot je že bilo rečeno, uporabnik program zažene v terminalu. Kar pomeni, da deluje na vseh operacijskih sistemih, ki imajo terminal. Ko uporabnik zažene program, se podatki, ki jih računalnik lahko pridobi iz bios-a vpišejo v program. Te prepoznamo po tem, da nimajo svetlo modrega ozadja. To nam prav tako nakazuje, da jih uporabnik nima možnosti spremeniti. Ostale podatke, torej podatke s svetlo modrim ozadjem, pa mora uporabnik prilagoditi sam.

Prvo stvar, ki jo lahko uporabnik spremeni je »OS active image«. To je gumb, ki odpre malo okno, kjer lahko uporabnik izbere verzijo A ali verzijo B. Do slednjega okna (Slika 6 Malo okno progama) dostopamo preko gumba »version«.



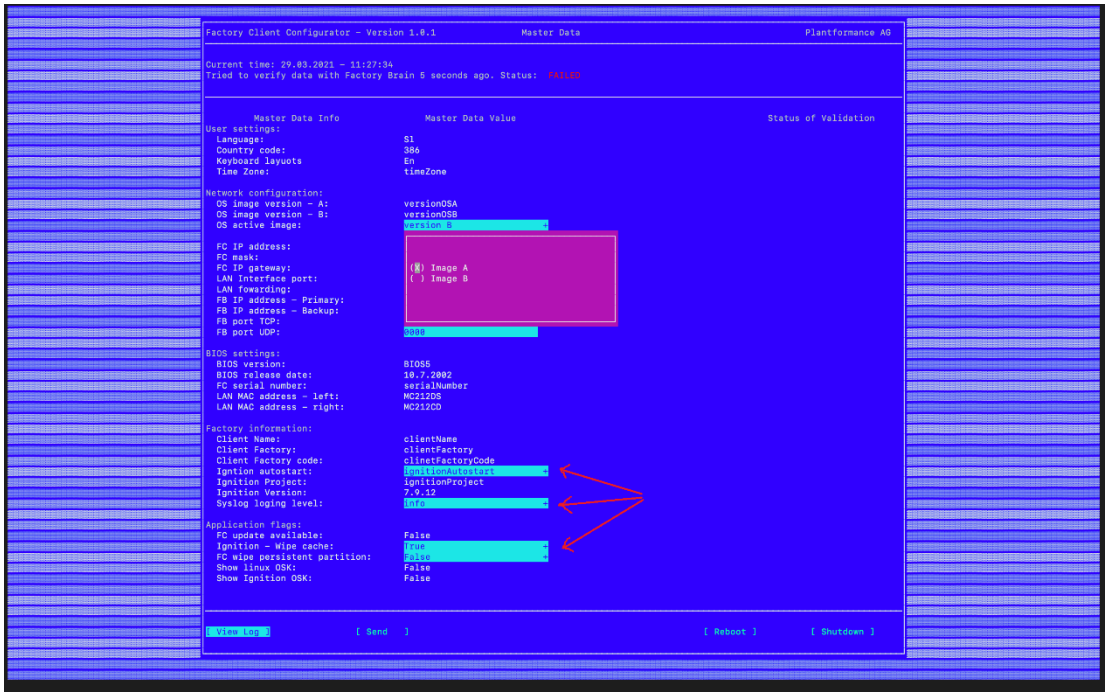
Slika 6 Malo okno progama

Naslednji prilagodljivi podatki so mrežni podatki (Slika 7 Podatki v programu), kot so IP naslovi, maske in vrata. Pri spremembi teh podatkov aplikacija poskrbi, da lahko uporabnik vpisuje podatke le v obliki številk. S tem preprečimo vpisovanje podatkov v neustreznem formatu.



Slika 7 Podatki v programu

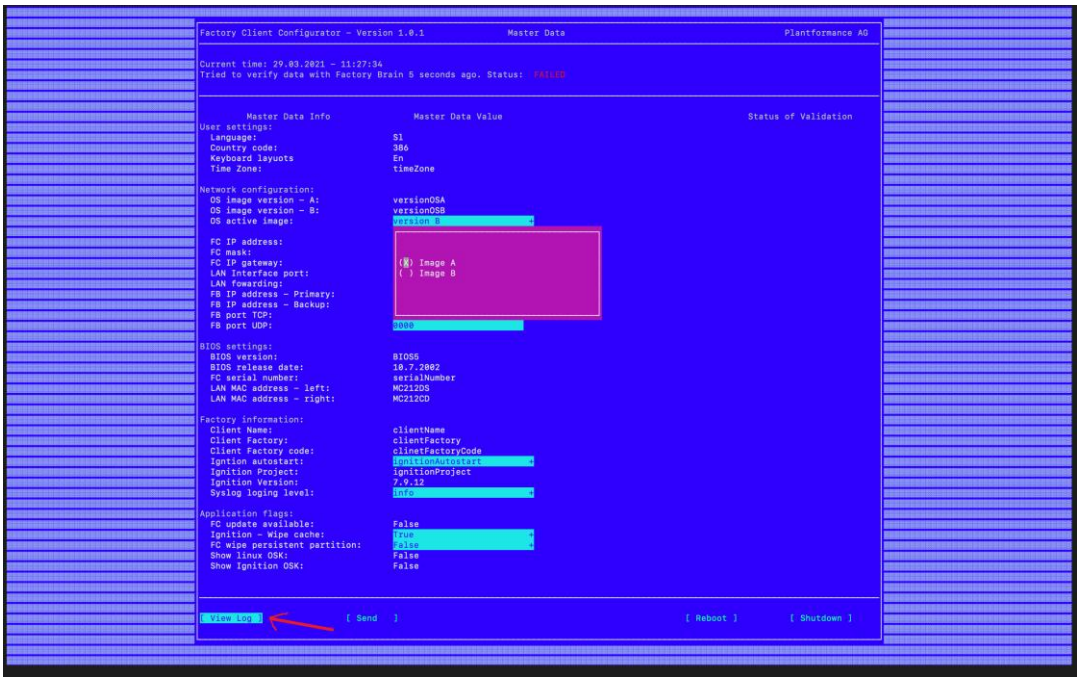
Še nižje imamo štiri tipke (Slika 8 pop-up gumbi), ki prav tako ob sprožitvi odprejo zavihek oziroma malo okno. Tipka »Ignition autostart« določa, če se bo program ob zagonu računalnika zagnal tudi sam. Gumb »Syslog loging level« nam določa filtre za dokumentiranje dogodkov. Pod te filtre spada »warning«, ki beleži opozorila, »info«, ki nam pokaže pozitivne scenarije, »err«, ki predstavlja napake in »debug«, ki aktivira vse naštete filtre.



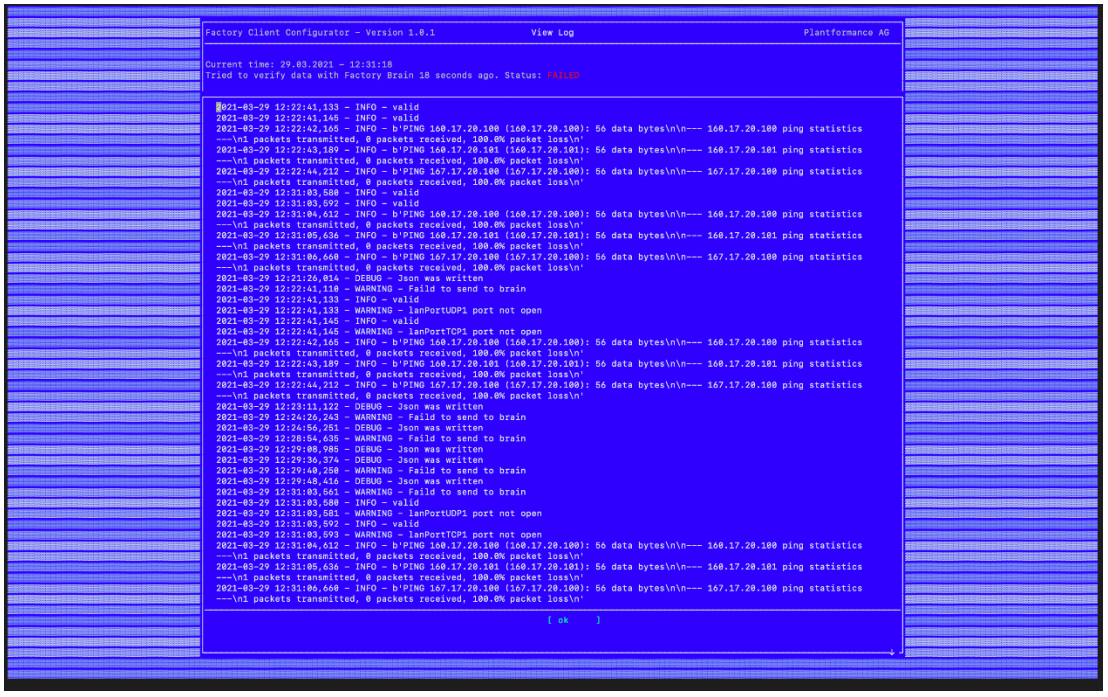
Slika 8 pop-up gumbi

Ko uporabnik določi vse podatke v telesu, ima v nogi programa štiri tipke, ki omogočajo delo s podatki, spremljanje posledic tega dela in pa manipulacijo z računalnikom.

Tipka »View log« (Slika 10 View Log tipka) nam odpre celozaslonsko okno (Slika 9 Celozaslonsko okno), v katerem so zabeleženi vsi dogodki glede na filtre, ki smo jih določili v telesu. Ima enako glavo kot glavni program, in nogo z eno tipko »ok«, ki zapre log okno.



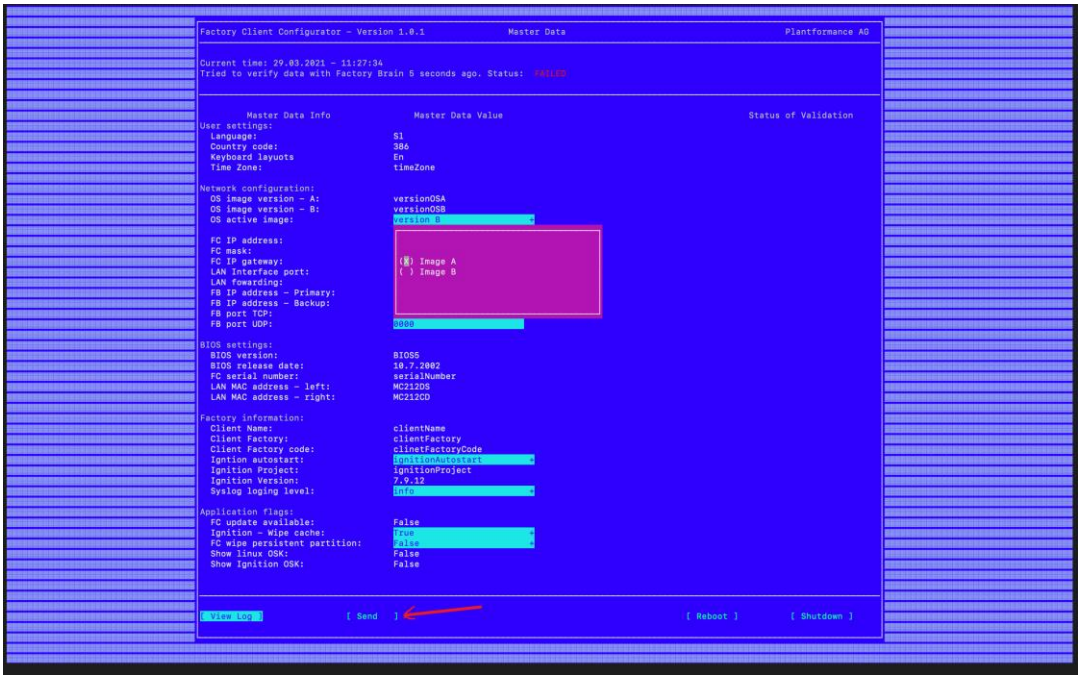
Slika 10 View Log tipka



Slika 9 Celozaslonsko okno

Naslednja tipa je tipka »Send« (Slika 11 Tipka "Send"), ki vse podatke v telesu pošlje na strežnik. Ti se naprej zakodirajo v base64. IP strežnika je uporabnik določil sam v telesu.

Sledita tipki »Reboot«, ki ponovno zažene računalnik in »Shutdown«, ki računalnik popolnoma ugasne.



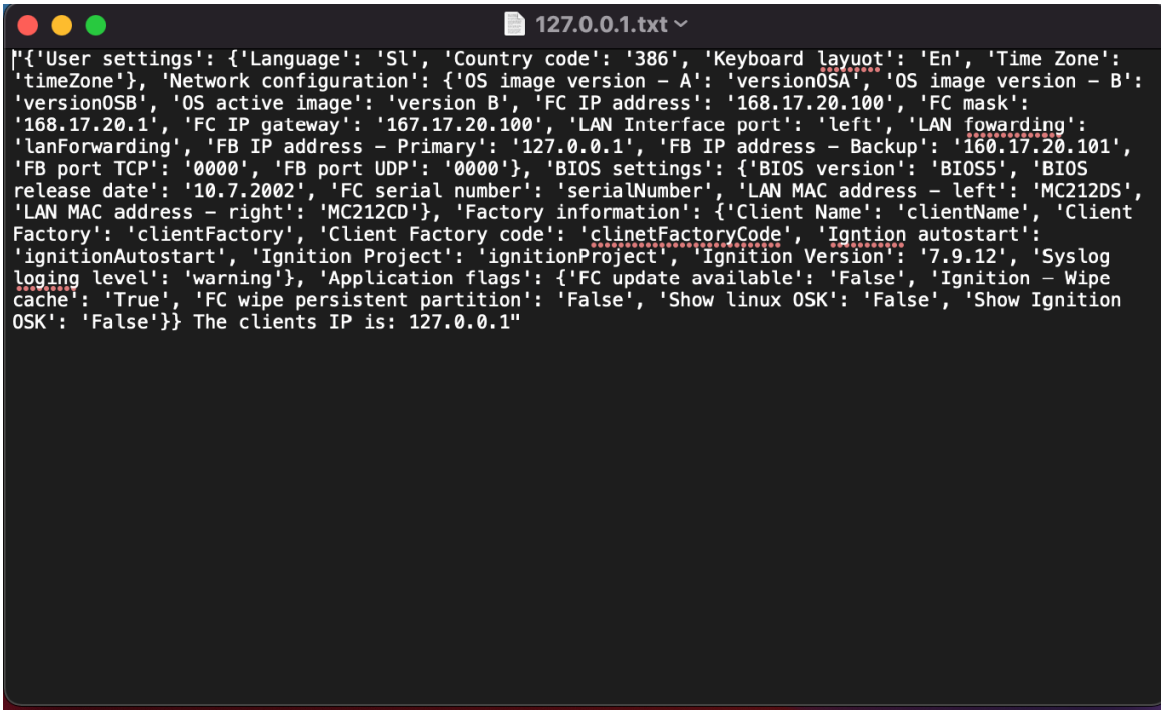
Slika 11 Tipka "Send"

4.7 Sekundarni program – Strežnik

Glavni program sam po sebi nima veliko vrednosti. Razlog za njegov obstoj je dokumentacija na strežniku, zato je edinole smiselno, da se implementira tudi sekundaren program – server. Ta vso besedilo, ki ga dobi od glavnega programa, shrani na računalnik v obliki datoteke (Slika 12 Datoteka serverja), na katerem je zagnan. Poimenuje ga po IP-ju pošiljatelja. Tako imamo zabeleženo, kateri podatki so od katerega računalnika.

Slika 12 Datoteka serverja

Ko strežnik dobi besedilo, je sprva kodirano v base64, zato ga ta dekodira. Shranitev dokumenta je možna tako v.txt kot .json formatu. Takšna formata sta prijazna za vse vrste računalnikov in prikazujeta podatke v človeku berljivi obliki (Slika 13 Podatki v datoteki).



Slika 13 Podatki v datoteki

5 ANALIZA

Predvideval sem, da splošni uporabniki računalniških sistemov ne bodo imeli veliko koristi od mojega programa, niti ni namenjen splošnim uporabnikom. Prav tako sem bil prepričan, da tudi če jim bi se jim zdel moj program zanimiv in uporaben, večina ljudi nima večjega števila računalnikov. Prav sem tudi predvideval, da bodo podjetja z več računalniki bolj zainteresirana v moj program. Moja hipoteza je bila, da več naprav kot jih imajo, bolj jim bo moj program olajšal delo.

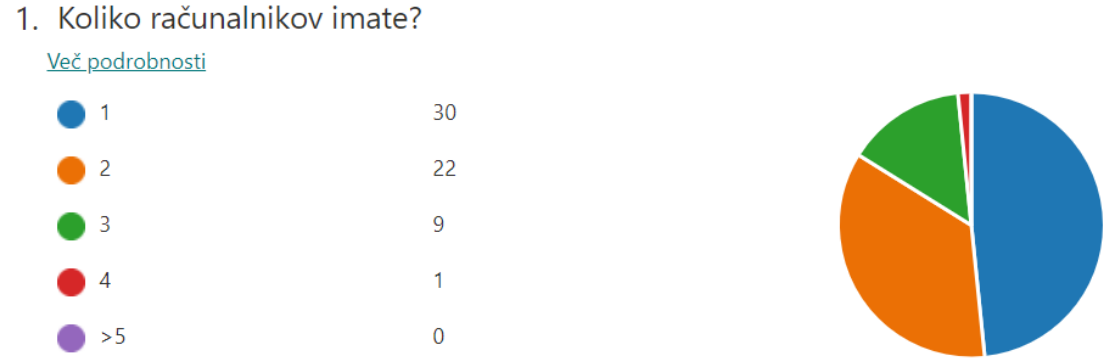
Za anketiranje sem naredil dve anketi, eno za navadne uporabnike računalnikov in eno za podjetja, ki se z računalniki ukvarjajo.

5.1 Splošni uporabnik

Kot je že bilo omenjeno sem predvideval, da splošni uporabniki ne bojo izkazali veliko zanimanja za moj program, menil sem tudi, da ne shranjujejo svojih podatkov na strežnikih.

Anketo, ki sem jo sestavil za splošne uporabnike računalnikov je rešilo 62 ljudi.

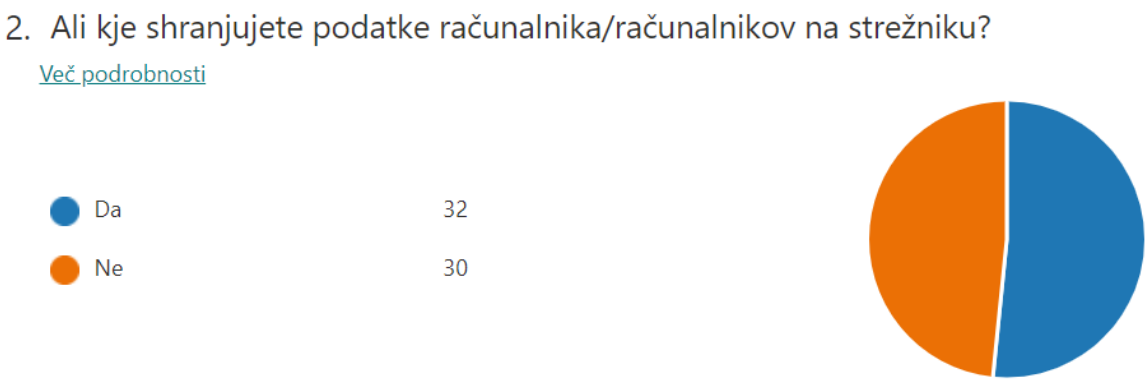
Prvo vprašanje, ki sem jim zadal, je bilo, koliko računalnikov imajo. Na kar je večina uporabnikov odgovorila, da imajo samo en računalnik. Na spodnji sliki (Slika 14 Prvo vprašanje - Uporabnik) so vidni tudi drugi odgovori, ki so prikazani na tortnem diagramu.



Slika 14 Prvo vprašanje - Uporabnik

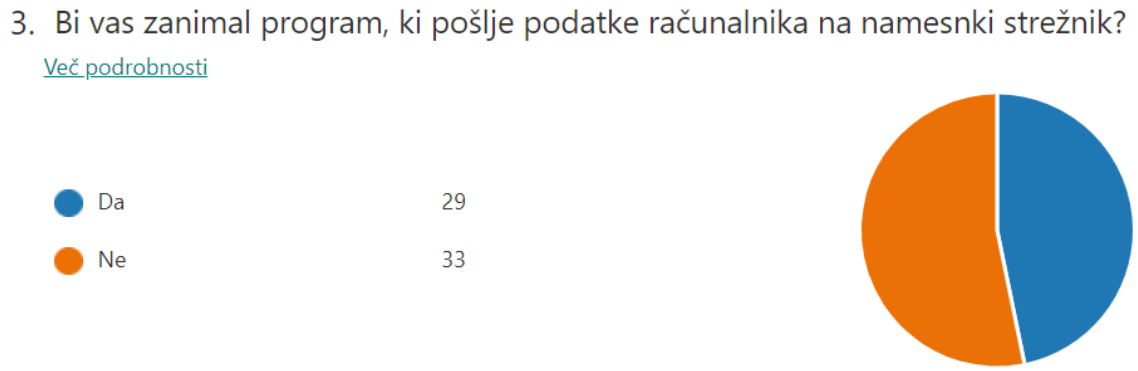
Drugo vprašanje. Ki sem ga postavil je bilo, če kje shranjujejo podatke svojega računalnika. Odgovori so bili zelo presenetljivi, saj je več kot polovica odgovorilo z da.

Na spodnji sliki (Slika 16 Drugo vprašanje - Uporabnik), lahko vidite, da je 32 uporabnikov odgovorilo z »da« in 30 z »ne«.



Slika 16 Drugo vprašanje - Uporabnik

Zadnje postavljeno vprašanje (Slika 15 Tretje vprašanje - Uporabnik) se je glasilo, če bi jih zanimal program, ki pošilja njihove podatke računalnika na strežnik.



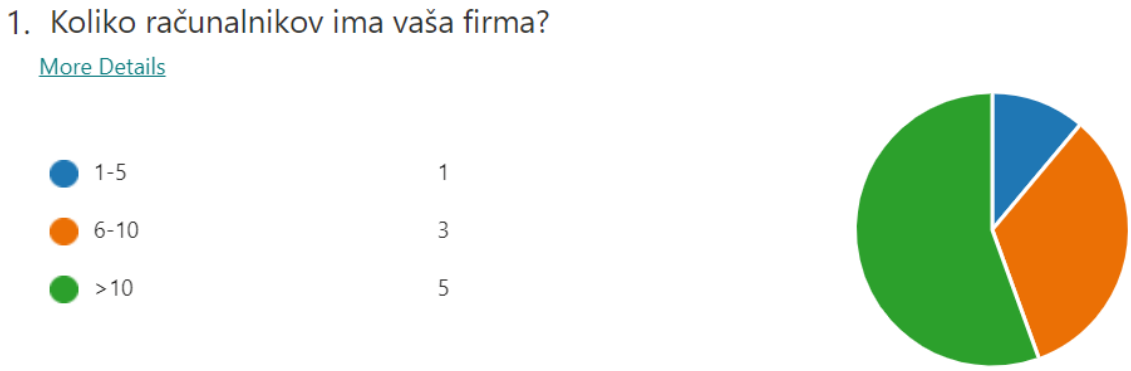
Slika 15 Tretje vprašanje - Uporabnik

5.2 Podjetja

Pri podjetjih sem pričakoval več zanimanja za svoj program, kot pri uporabnikih, ampak ni bilo tako.

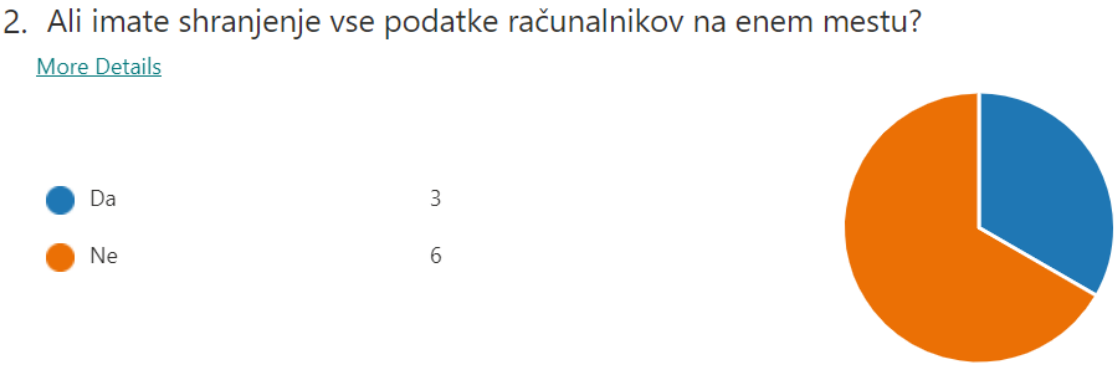
Uspelo mi je anketirati samo 8 podjetij, ker so samo ta podjetja bila pripravljena rešiti anketo.

Prvo vprašanje (Slika 17 Prvo vprašanje - Podjetja), ki sem ga zastavil, je bilo, koliko računalnikov imajo v podjetju. Odgovori niso bili presenetljivi, saj sem pričakoval, da bo večina podjetij imela več kot 10 računalnikov.



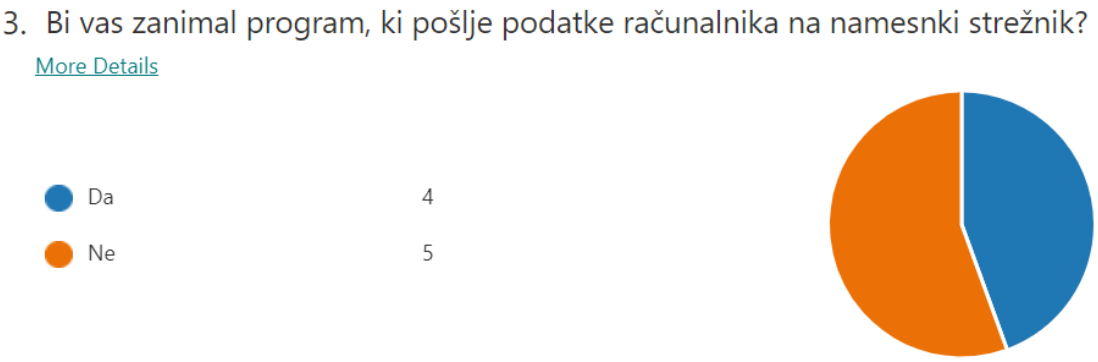
Slika 17 Prvo vprašanje - Podjetja

Pri drugem vprašanje (Slika 18 Drugo vprašanje - Podjetja) me je zanimalo, če imajo vse podatke shranjene na enem mestu. Pričakoval sem, da bo večina odgovorov da, ampak je 6 od 9 podjetij odgovorilo z ne.



Slika 18 Drugo vprašanje - Podjetja

Tretje vprašanje (Slika 19 Tretje vprašanje - Podjetja) je povpraševalo, ali bi jih zanimal program, ki pošlje na strežnik podatke njihovih računalnikov. Pričakoval sem več zanimanja za takšen program, res pa je, da je specifičen in ga ne potrebujejo vsa podjetja.



Slika 19 Tretje vprašanje - Podjetja

Pri zadnjem vprašanju za podjetja (Slika 20 Četrto vprašanje - Podjetja) sem vprašal, kako sedaj shranjujejo podatke. Največ odgovorov je bilo Google Drive in OneDrive

4. Na kakšen način spremljate podatke vaših računalnikov sedaj?

9 Responses

ID ↑	Name	Responses
1	anonymous	Gdrive
2	anonymous	Nerazumljivo vprašanje
3	anonymous	Google drive
4	anonymous	dropbox
5	anonymous	drive
6	anonymous	onedrive
7	anonymous	google drive
8	anonymous	OneDrive
9	anonymous	one drive

Slika 20 Četrto vprašanje - Podjetja

5.3 Analiza hipoteze

Pri splošnih uporabnikih sem si zadal hipotezo, da ne shranjuje svojih podatkov na strežnike. To hipotezo moram zavrnit, saj je večina anketirancev odgovorilo, da svoje podatke shranjujejo na strežniku. Razlog za to so verjetno moderne oblačne storitve, ki olajšajo nadzor nad podatki s tem, da jih kopičijo na eno mesto.

Za splošne uporabnik sem imel tudi hipotezo, da večina nima večjega števila računalnikov. Kljub presenetljivem številu ljudi z dvema računalnikoma, je še vedno skoraj 50% ljudi, ki imajo v lasti samo en računalnik. To hipotezo lahko zaradi tega potrdim. Tudi, če bi večina

anketirancev odgovorila, da ima dva ali pa celo tri računalnike v osebni rabi, bi hipotezo potrdil, saj je to še vedno manjše število računalnikov in jih je zato veliko lažje upravljati.

Mislil sem, da bo večina podjetij imela zanimanje za moj program. Zaradi pomanjkanja podatkov in malega števila anketiranih podjetij, hipotezo težko potrdim ali ovržem. Tudi razlika med pridobljenimi odgovori ni velika – okoli 55% odgovor ni v prid moji hipotezi. Z večjim številom anketiranih podjetij bi se tehtnica lahko prevagala v meni bolj pozitivno smer.

Mislil sem tudi, da ima večina podjetji več kot 10 računalnikov. To hipotezo lahko potrdim, saj je večina podjetji odgovorilo, da ima več kot 10 računalnikov – približno 55%. In čeprav ni bilo število anketiranih podjetji majhno, dvomim, da bi se ta številka kaj bistveno spremenila.

6 ZAKLJUČEK

Shranjevanje podatkov na lahek in varen način je danes zelo pomembno. Podatki in informacije so danes izjemno vredne surovine. Tako izgubljeni kot ukradeni podatki lahko povzročijo veliko škodo posamezniku in obsežni organizaciji.

Prednost moje rešitve je, da se podatki dokumentirajo, kar prepreči izgubo podatkov, hkrati pa so tudi bolj varni, saj sem poskrbel za lastni strežnik. Tako ni skrbi, da bi podatki bili poslani neznanemu strežniku.

Za tiste podatke, za katere nam je mar, bomo vedno poskrbeli. Slabost moje rešitve je to, da imajo različni podatki različno vrednost pri različnih ljudeh. Kar pomeni, da so pri podjetjih lahko mrežni in strojni podatki računalnikov in njihova dokumentacija pomembne, za navadnega posameznika pa sploh ne. Posameznikom je načeloma zanimivejša dokumentacija in varnost bolj osebnih podatkov. Tako moji aplikaciji primanjkuje prilagoditev poslanih podatkov.

Rešitev bi lahko še nadgrajeval. Omogočil bi pošiljanje raznolikih podatkov, aplikacijo bi razširil na ljudem bolj prijazen vmesnik kot je terminal in na koncu bi poskrbel za še dodatno varnost. Tu bi lahko nadgradil zaščito komunikacije med odjemalcem in strežnikom, dodal enkripcijo in vpeljal overjanje uporabnika, ki želi dostopati do podatkov.

VIRI

Byrnes, K. (8. februar 2019). *4 Reasons IT Documentation is the Key to Your Success*. Pridobljeno iz ITGlue: https://www.itglue.com/blog/4-reasons-documentation-is-the-key-to-your-success/

Dixit, S. (30. julij 2019). *Beginners Guide to Client Server Communication*. Pridobljeno iz medium: https://medium.com/@subhangdxt/beginners-guide-to-client-server-communication-8099cf0ac3af

Foundation, Python Software. (16. april 2021). *General Python FAQ*. Pridobljeno iz Python: https://docs.Python.org/3/faq/general.html#what-is-Python

National Federation of Self Employed & Small Businesses Limited. (5. oktober 2019). *Why is data protection so important?* Pridobljeno iz fsb: https://www.fsb.org.uk/resources-page/why-is-data-protection-so-important.html

Ward, I. (2014). *Documentation*. Pridobljeno iz urwid: http://urwid.org/