TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

Reemet Kampus 134224IASB

TELEKOMMUNIKATSIOONISEADMETE TÖÖKINDLUSE TESTIMISE TEMPERATUURIKAMBRI JUHTIMISE KASUTAJALIIDES

Bakalaureusetöö

Juhendaja: Tarmo Robal
PhD

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Reemet Kampus

15.05.2017

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks oli arendada Ericsson Eesti AS Tallinna tehasele tarkvarapõhine graafiline kasutajaliides juhtimaks modifitseeritud temperatuurikambri tööd, pärida sellelt andmeid, neid töödelda ning võimaldama kasutajal suhelda ka iseseisva seadmega. Temperatuurikambri juhtimise kasutajaliides sai arendatud Microsoft Windows keskkonda kasutades C# programmeerimiskeelt ning Microsoft Visual Studio 2015 arenduskeskkonda.

Temperatuurikamber on seade, mille üheks osaks on kamber, mille temperatuuri on võimalik muuta, ning mida kasutatakse erinevate seadmete kvaliteedikontrolli läbiviimiseks. Käesolevas töös kirjeldatud temperatuurikamber on mõeldud telekommunikatsiooniseadmete kvaliteedikontrolli läbi viimiseks.

Töös tutvustatakse modifitseeritud *Rohde & Schwarz T-Line RF shielded Temperature Chamber* temperatuurikambrit ning sellega suhtlemist. Autor tutvustab arendatud tarkvara arhitektuuri kolmekihilise arhitektuuri näitel. Samuti tuuakse välja tarkvara tasemete arhitektuur. Autor esitab funktsionaalsed ning mitte-funktsionaalsed nõuded, mille põhjal tarkvara sai arendatud, ja nende analüüsi ning kirjeldab valminud temperatuurikambri juhtimise kasutajaliidest.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 5 peatükki, 17 joonist, 1 tabel.

Abstract

Control User Interface for Telecommunication Equipment Reliability Testing Temperature Chamber

The objective of this bachelor thesis was to develop software based graphical user interface for Ericsson Supply Site in Tallinn for controlling modified Rohde & Schwarz T-Line RF shielded Temperature Chamber which would also collect data and process data from the chamber and would also provide an option to communicate with an external device.

Temperature chamber is a device which consists of a chamber with controllable temperature and is used to test quality of various devices. Temperature chamber described in this thesis is meant for performing quality control of telecommunication devices.

In this thesis the temperature chamber characteristics and communication with it are described. The author describes the architecture of the developed software are by using three layered architecture model. The tiered architecture of the software is also described by the author. The author presents and analyses the functional and non-functional requirements for developing the control user interface for the temperature chamber and describes the developed software by bringing out its different aspects.

The developed user interface for the temperature chamber control was able to deliver required functionality to the end user and thus was deployed into work by Ericsson Production Plant in Tallinn. Ericsson Production Plant team successfully took over the administration of the developed software.

The thesis is in Estonian and contains 33 pages of text, 5 chapters, 17 figures, 1 table.

Lühendite ja mõistete sõnastik

DUT	*Device Under Test*, testitav seade
Jäme klient	Klient, mis tegeleb suure osaga programmi andmetöötlusest
Klient	Funktsionaalüksus, mis saab teenuseid serverilt
PLC	*Programmable Logic Controller*, programmeeritav loogika kontroller
RF	*Radio Frequency*, raadiosagedus
String	Tervikuna käsitletav elementide järjend

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Antud bakalaureusetöö ülesandeks oli arendada tarkvaraline rakendus Ericsson Eesti Tallinna tehasele, mis võimaldaks kasutajal juhtida temperatuurikambri tööd, koguda ning jäädvustada kambrilt saadavaid andmeid ja suhelda ühe iseseisva seadmega läbi graafilise kasutajaliidese Microsoft Windows keskkonnas.

Temperatuurikambreid kasutatakse mitmetes erinevates tööstusvaldkondades, kus on oluline viia läbi toote kvaliteedikontroll enne lõppkasutajani jõudmist. Telekommunikatsiooniseadmete tootmine on üks sellistest valdkondadest. Kvaliteedikontrolli käigus jälgitakse toote käitumist erinevatel tingimustel, mis võivad aset leida lõppkasutaja käes. Temperatuurikambrid võimaldavad jälgida toote käitumist erinevatel temperatuuridel ning määrata ära toote kasutamise äärmustingimused, mida ületades ei suuda toode pakkuda tootja poolt pakutavat funktsionaalsust.

Antud bakalaureusetöös kirjeldatakse Ericsson Eesti Tallinna tehasele arendatud programmi, mis on mõeldud pakkumaks graafilist kasutajaliidest ja töö juhtimise võimalust modifitseeritud *Rohde & Schwarz T-Line RF Shielded Temperature Chamber* temperatuurikambrile ning iseseisvale seadmele. Eelnimetatud temperatuurikambri testsüsteem oli muutunud kasutuskõlbmatuks, mistõttu ei olnud enam võimalik temperatuurikambri tööd juhtida. Temperatuurikambri töö juhtimiseks ühendati sellega personaalarvuti, millele oli vaja arendada uus kasutajaliides ning suhtluse osa.

Temperatuurikambri suhtlusele ning kasutajaliidesele püstitati esialgsed funktsionaalsed ning mitte-funktsionaalsed autori poolt koostöös Ericsson Eesti Tallinna tehase tootetehnoloogia osakonna meeskonnaga. Tarkvara arendamisel oli prioriteediks aeg, mistõttu oodati autorilt võimalikult kiiresti vastavalt nõuetele toimiva programmi loomist.

Käesoleva töö peatükis 2 tutvustatakse temperatuurikambrit, mida töös kasutati, ning sellega suhtlemist. Kolmandas peatükis tutvustatakse arendatud tarkvara arhitektuuri, ning tuuakse välja nõuded, mille alusel arendati töös käsitletav programm. Töö viiendas peatükis, kirjeldatakse arendatud tarkvara ennast.

2 Temperatuurikamber telekommunikatsiooniseadmete töökindluse testimiseks

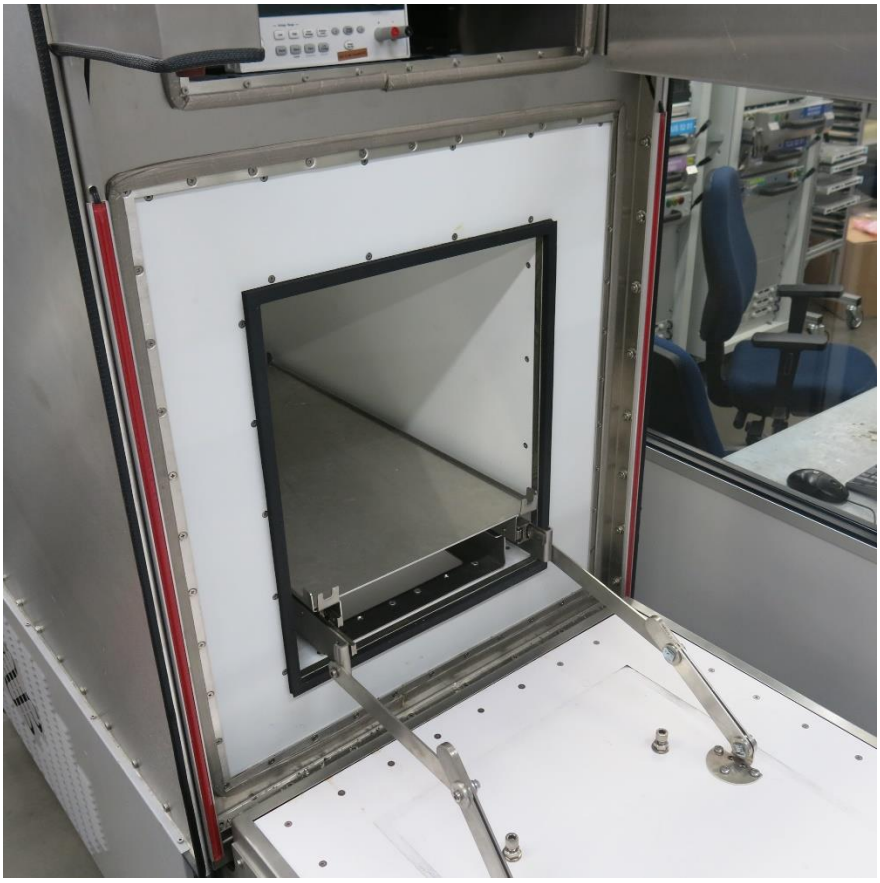
Temperatuurikamber, millele on käesolevas töös kirjeldatav tarkvara arendatud, on modifitseeritud *Rohde & Schwarz T-Line RF Shielded Temperature Chamber*. Temperatuurikamber on mõeldud telekommunikatsiooniseadmete töökindluse testimiseks temperatuuridel −40 °C kuni 80 °C.

Modifitseeritud temperatuurikamber (Joonis 1, Joonis 2) koosneb Rohde & Schwarzi varjestatud raamistikust, pneumaatilise uksega RF (*Radio Frequency*) kambrist, Franconia automatiseeritud külmutus ja kuumutus seadmetest ja Siemens PLC (*Programmable Logic Controller*) kontrollerist [1]. Läbi kontrolleri on võimalik muuta kambri temperatuuri ning sulgeda, lukustada ja avada kambri pneumaatilist ust.

Temperatuurikambril, millele kasutajaliides arendati, oli kunagi testsüsteem koos Rohde & Schwarzi tarkvaraga, kuid see oli lakanud töötamast. Testsüsteem, mis kunagi pakkus võimalust juhtida temperatuurikambri tööd, sooritada mõõtmisi ning juhtida testitavaid seadmeid, on tänaseks tehnoloogiliselt iganenud, sellel puudub kaabeldus ning see oli mõeldud ainult raadio- ja kõnesageduse seadmete jaoks, mille tõttu leiti, et selle parandamine ei ole mõtekas. Testsüsteem asendati personaalarvutiga, millel oli installeeritud *Microsoft Windows 7 Enterprise* operatsioonisüsteem. Sellele arvutile sai arendatud kasutajaliides.

Suhtluse võimaldamiseks arvuti ja temperatuurikambriga on vajalik ühendus arvuti jadapordi ning Siemens PLC kontrolleri jadapordi vahel. Ühenduse loomiseks kasutatakse RS232C standardile vastavat kaablit DB9 konnektoritega.

Joonis 1. Temperatuurikamber.



Joonis 2. Temperatuurikamber avatud RF kambri uksega.

Suhtlemine arvuti ja kontrolleri vahel toimub asünkroonse järjestikedastusena. Asünkroonne suhtlus tähendab, et seadmed ei vaheta omavahel andmeid, et teavitada teineteist millises olekus nad hetkel on [2]. Suhtluse korrektseks toimimiseks peavad mõlemad seadmed edastama sõnumeid modulatsioonikiirusega 9600 boodi ja kasutama kaheksat andmebiti [1]. Andmete edastamiseks saadab üks seade start biti, seejärel ülejäänud andme bitid ning lõpetab stopp bitiga. Paarsusebitti suhtluses ei kasutata.

Kontroller reageerib ainult kindlatele sõnumitele. Neid sõnumeid on kokku kümme, kuid töös kirjeldatav programm kasutab neist ainult nelja (Tabel 1).

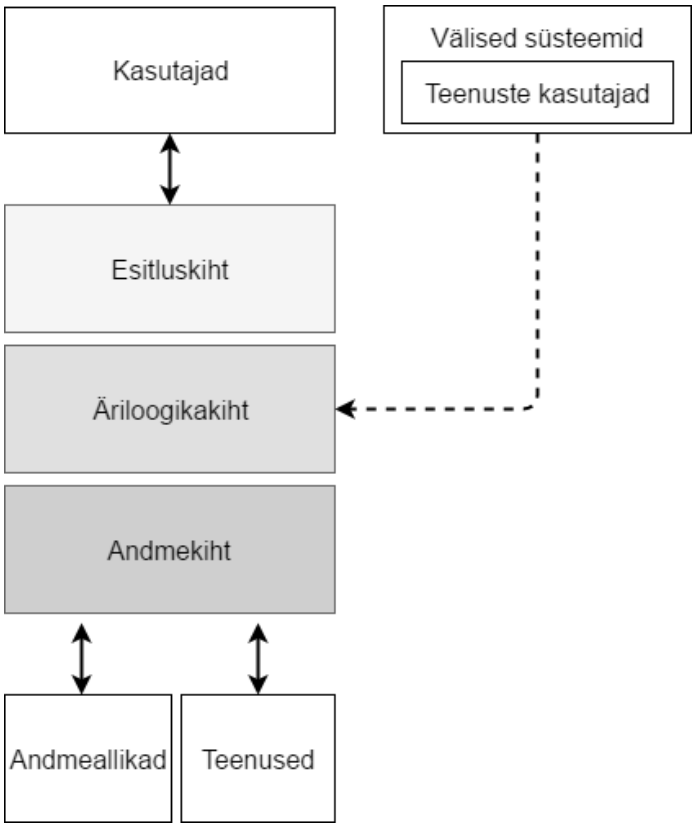
Tabel 1. Kontrolleri käsud [1].

Käsk kontrollerile	Vastus kontrollerilt	Kirjeldus
SYST:DOOR REL<CR>		Kambri ukse lahti lukustamine
TEMP:SET xx.x <CR>		Kambri temperatuuri määramine
TEMP? <CR>	Näiteks: „23.5“	Küsitakse kambri temperatuuri
TEMP:CL?<CR>	Näiteks: „23.5“	Küsitakse jahutusvedeliku temperatuuri

3 Temperatuurikambri kasutajaliidese ja juhtimise arhitektuur ning nõuded süsteemile

Kõikide arvutiprogrammide arhitektuuri on võimalik kirjeldada kihtide ja tasemetega. Kihiline arhitektuur kujutab endast programmi funktsionaalsuse loogilist grupeerimist eri osadeks [3]. Süsteemi jaotamine kihtideks aitab arendajal mõista üksikut kihti sidusa tervikuna ilma eelduseta, et mõistetakse süsteemi teisi kihte. Samuti võimaldab jaotus arendajal asendada kihte alternatiivsete baasteenuste rakendamisega ja minimaliseerida sõltuvust eri kihtide vahel [4]. Süsteemi jaotamine kihtideks võib luua ka probleeme, kuna alati ei ole võimalik kõiki süsteemi osi selgelt eraldada eri kihtidesse, mille tõttu muudatused ühes kihis mõjutavad teisi kihte, mida peab seetõttu hakkama muutma ning need muudatused mõjutavad omakorda veel järgmisi kihte. Samuti võivad kihid põhjustada langust süsteemi jõudluses, näiteks tingituna andmete muutmisest vastava kihi vormingusse [4].

Programmi arhitektuuri saab väljendada kolme põhilise kihi abil: esitluskiht, äriloogikakiht ja andmekiht (Joonis 3). Esitluskiht käsitleb kasutaja ja tarkvara vahelist suhtlust. Esitluskihi põhiülesanneteks on informatsiooni kuvamine kasutajale ning tema käskude tõlgendamine ülesanneteks äriloogika- ja andmekihile. Vastavalt esitluskihist saadud käskudele, tegeleb äriloogikakiht sealt tulnud andmete valideerimisega, andmekihi rakendamisega ja arvutustega, mis tulenevad sisestatud või salvestatud informatsioonist [4]. Andmekiht suhtleb süsteemidega, mis täidavad ülesandeid programmi jaoks. Andmekiht võimaldab ka juurdepääsu andmetele, mida hoiustatakse süsteemis ning mis pärinevad välistest süsteemidest ja/või võrkudest [3].



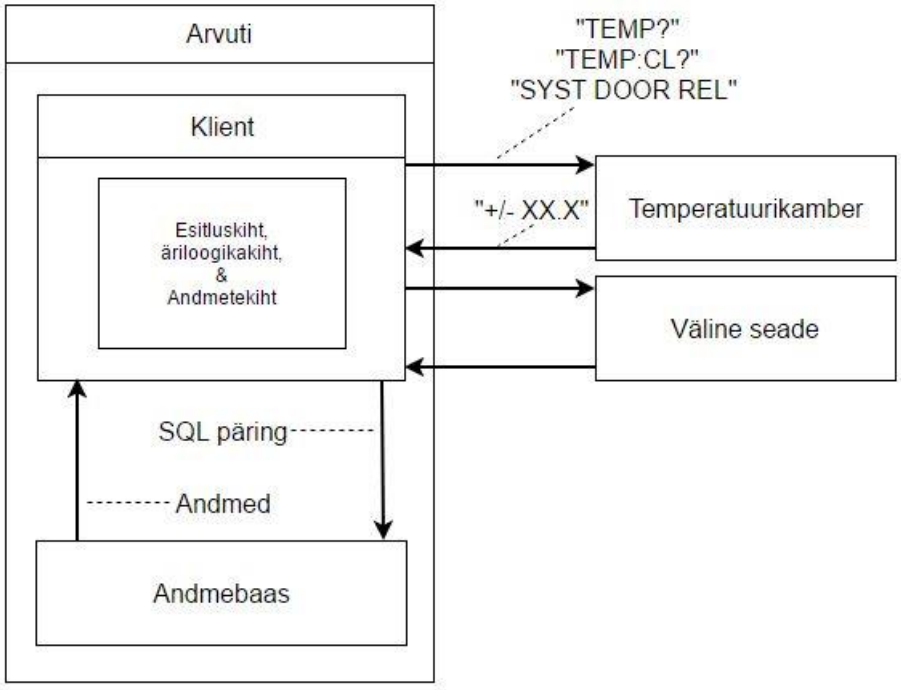
Joonis 3. Lihtsustatud kihiline arhitektuur [3].

Arhitektuuri tasemed kujutavad endast programmi funktsionaalsuse ja komponentide füüsilist jaotamist arvutite, serverite, võrkude või eraldatud seadmete vahel. Jaotatakse ka kolme eelnevalt mainitud kihti: esitluskiht, äriloogikakiht ja andmekiht. Kihtide jaotusmustrist oleneb kuidas sellist jaotust nimetatakse. Põhjusi miks jagada programmi funktsionaalsust ja komponente eri seadmete vahel on mitmeid, näiteks välistatakse andmebaasi paigutamisel serverisse vajadus sünkroniseerida teavet erinevate seadmete vahel, kui kõik võrgus olevad seadmed kasutavad serveris olevat andmebaasi. Samuti lihtsustab tasemeteks jagamine arendaja tööd, kui soovitakse olemasolevat programmi täiustada, kuna ühte taset on võimalik muuta ilma teisi tasemeid segamata.

Käesolevas töös kirjeldatud süsteem koosneb jämedast kliendist ja SQLite andmebaasist (Joonis 4). Jämedaks kliendiks nimetatakse klienti, mis tegeleb suure osaga programmi andmetöötlusest. Tegemist on kohaliku SQLite andmebaasiga, mis asub kliendiga samas personaalarvutis, kuid on rakendusest eraldiseisev ja seega saab seda muuta ilma klientprogrammi muutmata. Kasutuses olev arvuti täidab nii kliendi kui ka serveri ülesannet. Seetõttu võib pidada kirjeldatud arhitektuuri teise taseme ehk klient-server arhitektuuriks. Andmebaas on valitud kohalik, kuna ainult üks klientprogramm vajab seda

andmebaasi enda tööks. Samuti suudab personaalarvuti, millel asub klient ja andmebaas, hallata kliendi poolt nõutavaid andmeid suurepäraselt, kuna neid on suhteliselt vähe (peatükk 4.2). Samuti on selline lahendus lihtne, mis võimaldas kiiret rakendamist.

Käesolevas töös kirjeldatava programmi arhitektuuri esitlus-, äriloogika- ning andmekihti ei ole üksteisest rangelt eraldatud. Arhitektuuride kihtide range mitte eraldamine tähendab siin, et kihtide funktsionaalsust pole jagatud erinevatesse klassidesse. Kihid on pigem eristatud erinevate protseduuridega. Selline arhitektuur põhjustab tugevaid sõltuvusi erinevate kihtide vahel. Kui üks kiht peaks ootamatult lakkama korrektselt toimimast, siis teevad seda ka temast sõltuvad kihid. Selline arhitektuuriline lahendus ei ole sobilik suurtele ning keerukatele programmidele, mis tegelevad paljude erinevate ülesannetega, sest nende muutmisel peab hakkama ka kõiki loodud sõltuvusi arvesse võtma. Seetõttu on sellise arhitektuuriga suuri programme keeruline hallata. Samas on käesolevas töös tegu väikese programmiga ja seetõttu ei ole selle haldamine keeruline. Arhitektuuri range eraldamise puudumine võimaldas autoril pakkuda lühema ajaga funktsionaalsuse valmimist, kuna seda oli lihtne rakendada. Ericssoni Tallinna tehase tootetehnoloogia meeskonna jaoks oli aeg oluline faktor, mille tõttu tundus autorile sellise arhitektuuri kasutamine õigustatud.



Joonis 4. Kasutajaliidese ja suhtluse arhitektuur.

Konkreetse programmi esitluskiht tegeleb erinevate andmete kuvamisega, mis pärinevad temperatuurikambrilt ja andmebaasi tabelitest. Samuti tõlgendab esitluskiht kasutaja interaktsiooni, näiteks kasutajaliideses nupule vajutamise, käskudeks äriloogika kihile. Äriloogikakiht käitub esitluskihist tulnud käskudele vastavalt ettenähtud korrale, milleks on tihti andmekihi rakendamine suhtluseks andmebaasi või temperatuurikambriga. Äriloogika kiht tegeleb ka automaatsete toimingutega, mis ei vaja esitluskihist pärinevaid käske, nagu andmekihi rakendamine kambri temperatuuri pärimiseks ning selle edasi andmiseks esitluskihile kuvamiseks. Andmekiht tegeleb temperatuurikambri ning iseseisva seadmega suhtlemisega ning nende andmete edastamisega äriloogikakihile ja esitluskihile. Samuti tegeleb andmekiht andmete sisestamisega andmebaasi ning nende pärimisega andmebaasist.

Tarkvara arendamisel on tähtis, et klient ja arendaja mõistavad tehtavat tööd üheselt. Kliendil on tavaliselt mingisugune ettekujutus lõpptulemusest, kuid tihti on kirjeldus sellest liiga üldine, mistõttu võib tarkvaraarendaja kliendi kirjeldustest teisiti aru saada. Eriarusaamad arendatavast tootest võivad viia töö ümbertegemiseni, mis lõppeb tihti kliendi jaoks lisatasudega ning toote valmimisaja pikenemisega. Vältimaks selliseid olukordi, kus arendatav tarkvara ei vasta kliendi ettekujutusele, on mõistlik kliendil koos arendajaga välja töötada nõuded arendatavale tarkvarale, et mõlemal osapoolel oleks selge ettekujutus tehtavast tööst.

Käesolevas töös ei olnud kõik nõuded välja töötatud arendusprotsessi algul vaid osad lisandusid töö käigus, kui eelmised olid rakendatud. Nõuded tarkvarale koostas autor vastavalt tootetehnoloogia meeskonna kirjeldustele mida süsteem peab tegema ja kuidas.

Nõuded tarkvarale võib jaotada kaheks: funktsionaalseteks ja mitte-funktsionaalseteks. Funktsionaalsed nõuded kirjeldavad toiminguid, mida süsteem peab tegema või pakkuma kasutajatele. Mitte-funktsionaalsed nõuded kirjeldavad kuidas süsteem peab toimima.

3.1 Funktsionaalsed nõuded

Esimest nõuet võib lugeda baasfunktsionaalsuseks, mille autor sõnastas järgmiselt:

1. Kasutajaliides peab võimaldama kasutajal määrata temperatuurikambrile temperatuure, mida see peab saavutama, kasutaja poolt valitud ajaperioodideks.

Järgnevad funktsionaalsed nõuded töötati välja paralleelselt mitte-funktsionaalsete nõuetega. Autorilt oodati kõigepealt esimese nõude täitmist töötava prototüübi näol. Kui prototüüp täitis seda ühte nõuet, hakati välja töötama uusi nõudeid. Järgmisena sooviti autorilt, et kasutaja saaks luua jada erinevatest temperatuuridest ja nende soovitud kestvusaegadest. Seejärel hakkaks kasutajaliides saatma loodud jadast temperatuure kambrile saavutamiseks ja neid hoitaks vastavalt nende kestvusaegadele. Kestvusaja möödumisel saadetakse jadast järgmine temperatuur, kamber saavutab selle ning seda temperatuuri hoitakse vastavalt selle kestvusajale ja nii kuni jada viimase temperatuurini. See kasutajalugu sai kaetud esimesele nõudele lisades teise ja kolmanda nõude:

2. Süsteem peab kuvama kasutaja poolt sisestatud temperatuurid ja kestvusajad.
3. Süsteem peab määrama temperatuurikambrile temperatuure vastavalt nende sisestusjärjekorrale tingimusel, et eelneva temperatuuri saavutamisest kambris on möödunud kestvusajaga sama aeg (välja arvatud kui tegemist on järjekorras esimese temperatuuriga).

Sellist jada, mis on koostatud temperatuuridest ja kestvusaegadest, nimetatakse edaspidi käsujadaks.

Klient ootab enamikul juhtudest arendatavalt programmilt ka kasutajasõbralikkust, seetõttu pakkus autor välja järgnevad nõuded:

4. Süsteem peab võimaldama kasutajal säilitada sisestatud temperatuurid koos vastavate ajaperioodidega. Säilitamise eesmärk on eemaldada vajadus programmi igal käivitusel uuesti määrata temperatuurikambrile samad temperatuurid, mida soovitakse saavutada.
5. Süsteem peab kuvama veateate, kui sisestatud temperatuur ei jää vahemikku −40°C kuni 80°C. Tegemist on temperatuurikambri poolt saavutavate äärmustemperatuuridega.
6. Süsteem peab kuvama veateate, kui sisestatud kestvusaeg on väiksem või võrdne nullist.

Järgnevad nõudeid tootetehnoloogia kollektiiv spetsiifiliselt ei maininud, kuid autor pakkus need välja. Need nõuded võivad tunduda nii elementaarsed, et neid ei tule pähegi välja pakkuda, mille tõttu nende mitte ülestähendamisel ei pruugi lõpptoode vastata kliendi ettekujutusele, kuna puudub osa funktsionaalsusest ja kasutajasõbralikkusest:

7. Kasutajal peab olema võimalus peatada süsteemil edasist temperatuuride määramist temperatuurikambrile.
8. Süsteem peab kuvama temperatuurikambrile viimati määratud temperatuuri koos selle ajaperioodiga.
9. Kasutaja peab saama kustuda säilitatud temperatuure koos neile vastavatele ajaperioodidega.

Kasutajaliideselt sooviti, et see kuvaks ja töötleks temperatuurikambrilt pärinevaid andmeid erinevatel viisidel:

10. Süsteem peab kuvama temperatuurikambri hetke temperatuuri ning seda uuendama automaatselt.
11. Süsteem peab kuvama graafikut, mis kuvab temperatuurikambri temperatuuri ajas. Lisaks temperatuurile peab graafik kuvama kambrile saadetud temperatuuri selle saatmise ajahetkel koos vastava ajaperioodiga.
12. Süsteem peab uuendama graafikut automaatselt 10 sekundilise intervalliga. Algselt oli tegemist 1 sekundilise intervalliga, kuid tootetehnoloogia meeskond pakkus välja 10 sekundilise intervalli, kuna temperatuuri muutumine kambris on suhteliselt aeglane.
13. Süsteem peab võimaldama kasutajal puhastada graafikut andmetest.
14. Süsteem peab võimaldama kasutajal graafikul kuvatavad andmed eksportida Microsoft Exceli formaati.
15. Süsteem peab kasutajat teavitama temperatuurikambri töövalmidusest. Kambrit peetakse töövalmis, kui jahutusvedeliku temperatuur on alla −20°C.

Viieteistkümnes nõue on üks hilisematest funktsionaalsetest nõuetest. See nõue sai lisatud peale teiste funktsionaalsete nõuete realiseerimist (ka peale neid, mis numbrilises järjestuses järgnevad viieteistkümnendale nõudele). Viieteistkümnes nõue tekkis temperatuurikambri töös täheldatud suurte temperatuuri kõikumiste jälgimise tulemusena.

Ericssoni tootetehnoloogia meeskond soovis, et läbi kasutajaliidese oleks võimalik ka suhelda veel temperatuurikambrist erineva seadmega. Seda põhjendusel, et nii oleks võimalik läbi ühe kasutajaliidese võimalik ka juhtida kambris olevat seadet ehk DUT (*Device Under Test*).

16. Kasutajaliides peab võimaldama kasutajal suhelda temperatuurikambrist sõltumatu seadmega.
17. Kasutajaliides peab kuvama iseseisvast seadmest tulevad andmed.
18. Kasutajaliides peab võimaldama kasutajal valida ajaperioodi, mille möödumisel saadetakse kasutaja poolt sisestatud käsk iseseisvale seadmele.
19. Kasutajal peab olema võimalus peatada käskude saatmine iseseisvale seadmele.

Autor pakkus välja, et kasutajaliidesega oleks võimalik muuta temperatuurikambriga ja iseseisva seadmega suhtlemiseks vajalikke sätteid. Nõuetena pani autor kirja need järgmiselt:

20. Süsteem peab kuvama olekuinfot, milleks on ühenduse olemasolu temperatuurikambriga ning iseseisva seadmega, kasutuses olevad jadapordid, boodi kiirused ning tõlgendused.
21. Süsteem peab võimaldama kasutajal muuta jadapordi ühenduste boodi kiirust ja tõlgendust ning valida temperatuurikambri ja iseseisva seadme jadapordi ühenduse. Paarsuse muutmist ei peetud vajalikuks, mistõttu selle valik puudub nagu ka selle väärtus.

Peale eelmiste nõuete funktsionaalsuse rakendamist prototüüpi, soovis tootetehnoloogia kollektiiv, et kasutajaliides looks sündmuste logi faili ja faili kuhu salvestatakse kambrile viimati määratud temperatuuri. Nõuded said püstitatud järgmiselt:

22. Süsteem peab võimaldama kasutajal valida kataloogi kuhu süsteem kuvab temperatuurikambri staatuse ning sündmuste logi.
23. Süsteem peab looma kasutaja poolt valitud kataloogi kausta koos failiga, kuhu salvestatakse temperatuurikambri staatus ehk kambri temperatuur, kambrile viimati määratud temperatuur ja selle temperatuuri hoidmise ajaperiood ning järele jäänud aeg järgmise temperatuuri määramiseni.
24. Süsteem peab looma kasutaja poolt valitud kataloogi veel teise faili, kuhu salvestatakse sündmuste logi, mis kuvab lisaks 23. punktis nimetatud atribuutidele veel salvestamise aja.

Põhjus miks sooviti, et süsteem koostaks logi ning omaks temperatuurikambri olekufaili, oli selles, et kasutajad saaksid läbi võrgu jälgida, mida kamber hetkel teeb (olekufail) ning mida ta siiani teinud on (logifail). Sellelt funktsionaalsuselt oodati lihtsust.

Viimane funktsionaalne nõue sai lisatud temperatuurikambriga töötamisel esinenud anomaalia tõttu. Meeskonna liikmed ei saanud kambri ust avada, sest see oli lukustunud teadmata põhjustel. Ukse lukustus oli võimalik avada saates temperatuuri kontrollerile vastava käsu. Ukse iseeneslikku lukustamist ei õnnestunud autoril ega ka tootetehnoloogia meeskonnal korrata, kuid olukorra lahendamise lihtsustamiseks tulevikus sai püstitatud järgnev nõue:

25. Kasutajaliides peab võimaldama kasutajal temperatuurikambri ukse lukku avada.

3.2 Mitte-funktsionaalsed nõuded

Mitte-funktsionaalsed nõuded said välja töötatud arendusprotsessi käigus ja paljude püstitamisega jäeti töö autorile vabad käed. Kõik nõuded esitati enne rakendamist ettevõtte meeskonnale ülevaatamiseks ja vajadusel kohendati neid.

Ericsson Tallinna tehase kollektiiv on eri rahvustest ja töökeelteks on eesti, vene ning inglise keel, millest viimane on rahvusvaheline suhtluskeel. Seetõttu sai esimene mittefunktsionaalne nõue kirja pandud järgmiselt:

1. Kasutajaliideses sisalduv tekst peab olema inglise keelne.

Graafikuga seoses nõuded tunduvad elementaarsed, kuid neid tuli siiski arusaamatuste vältimiseks üles märkida:

2. Sisestatud temperatuur peab olema Celsiuse skaalal ja selle hoidmise kestvus minutites.
3. Graafik peab koosnema aja ja temperatuuri teljest.
4. Graafikul peavad kambrile saadetud temperatuurid olema kuvatud annotatsioonina.
5. Graafiku teljed peavad olema reguleeritavad klaviatuuri ja/või hiirega.
6. Graafiku temperatuuri telg peab olema reguleeritav vahemikus −40 ℃ kuni 80 ℃ ning aja telg peab olema reguleeritav terve graafikule kantud andmete ulatuses.

Neljas nõue tekkis teistest graafikuga seotud nõuetest hiljem ja tekkis kasutusele võetud süsteemi piirangutest. Graafikul navigeerimine oli nõutud ettevõtte poolt (viies nõue), autor lisas sellele täpsustuse kuuenda nõude näol.

Ettevõte soovis, et programmi kasutaja saaks aru, mis failidega ning andmetega on tegemist ning et need oleksid vastavuses andmetega graafikul. Autor pakkus välja järgnevad mitte-funktsionaalsed nõuded:

7. Süsteemi poolt loodud kaust peab kandma nime „Status and history“.
8. Süsteemi poolt loodud failid kaustas „Status and history“ peab olema .txt formaadis ning kandma nime „Status“ ja „TempLog [loomise kuupäev ja aeg] “.
9. Faili „Status“ sisu peab kuvama andmeid formaadis: „Current step:[ ], time left: [ ] , current temp:[ ], time: [ ]“.
10. Faili „Status“ peab uuendama graafikuga samal ajal.
11. Faili „Status“ sisu pean igal uuendamisel üle kirjutama.
12. Faili „TempLog [loomise kuupäev ja aeg]“ sisu peab kuvama andmeid formaadis: „Current step:[ ], time left: [ ] , current temp:[ ], time: [ ]s“. Loomise kuupäev ja aeg pannakse faili nimesse, et eristada erinevaid teste.
13. Faili „TempLog [loomise kuupäev ja aeg]“ peab uuendama graafikuga samal ajal.
14. Faili „TempLog [loomise kuupäev ja aeg]“ uuendamisel peab iga sissekanne olema kirjutatud uuele reale.

Kaheksandas nõudes sai .txt formaat valitud selle rakendamise lihtsuse poolest, olles samas piisav, et esitada meeskonna poolt soovitud andmeid. Üheteistkümnendas nõudes nõutakse ülekirjutamist, et kasutajal oleks võimalikult kiirelt võimalik saada ülevaade temperatuurikambri töö hetkeseisust.

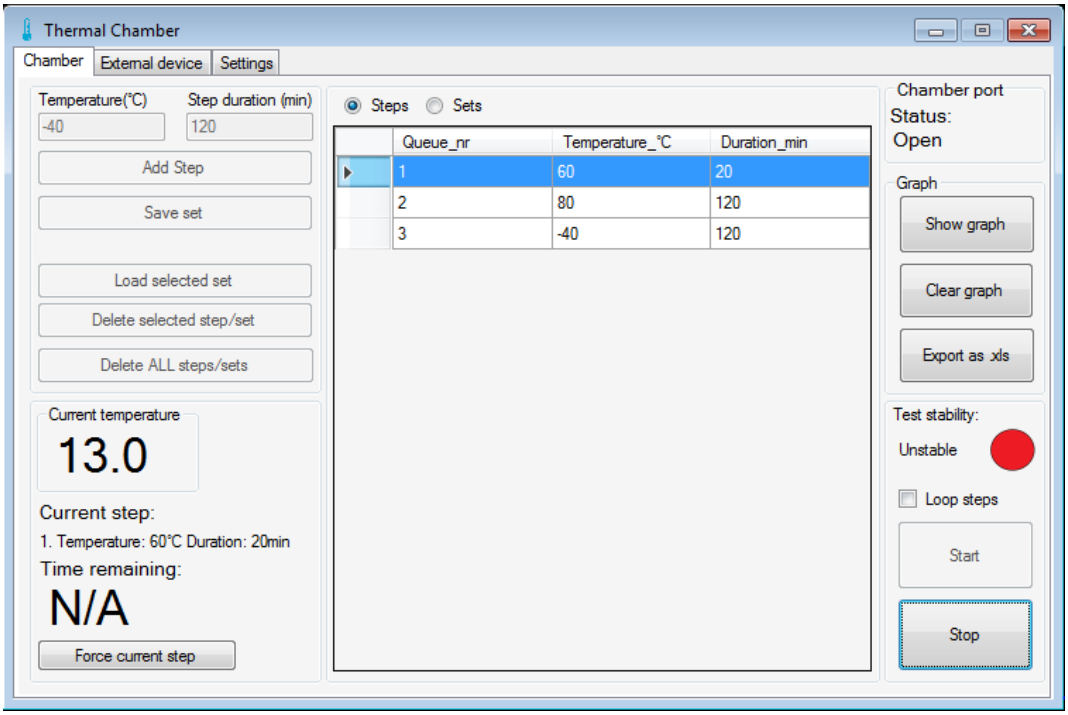
Nõuded iseseisva seadme suhtlusele olid lihtsad. Kollektiiv soovis, et iseseisva seadmega oleks võimalik suhelda läbi lihtsa terminali.

15. Iseseisva seadmega suhtlemine peab toimuma läbi terminali.
16. Iseseisvale seadmele peab saama saata käske kasutaja poolt manuaalselt või automaatselt kasutaja poolt sisestatud ajaperioodi tagant.

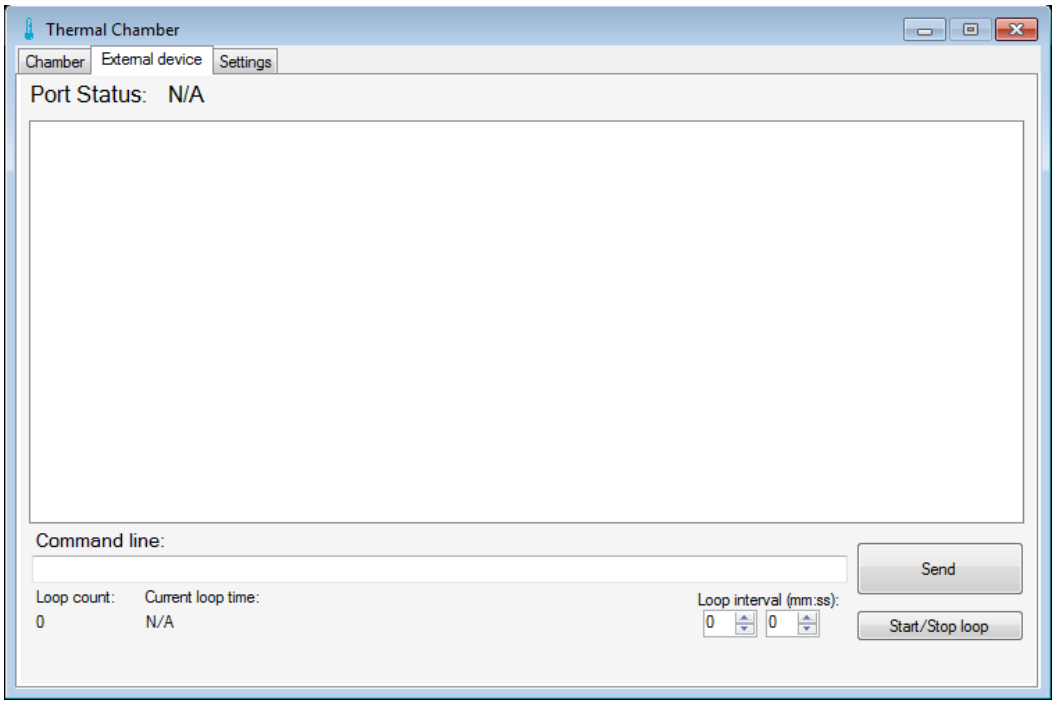
4 Temperatuurikambri juhtimise kasutajaliides

Kasutajaliides on arendatud programmeerimiskeeles C# kasutades Microsoft Visual Studio 2015 arenduskeskkonda, kuna nii keel kui ka keskkond on Ericsson Tallinna tehases tarkvaraarenduse standarditeks. Programm on suunatud Microsoft Windows platvormile, toetades nii 64 kui ka 32 bitist versiooni. Programmi korrektseks töötamiseks on vaja kahtteist teeki: System, System.Data, System.Drawing, System.Text, System.Windows.Forms, System.IO, System.IO.Ports, System.Globalization, Oxyplot, Oxyplot.WindowsForms, Excel = Microsoft.Office.Interop.Excel. Lisaks vajalikele teekidele on programmi tööks olulised veel kolm pilti: Red.png, green.png ja TempIcon1.ico.

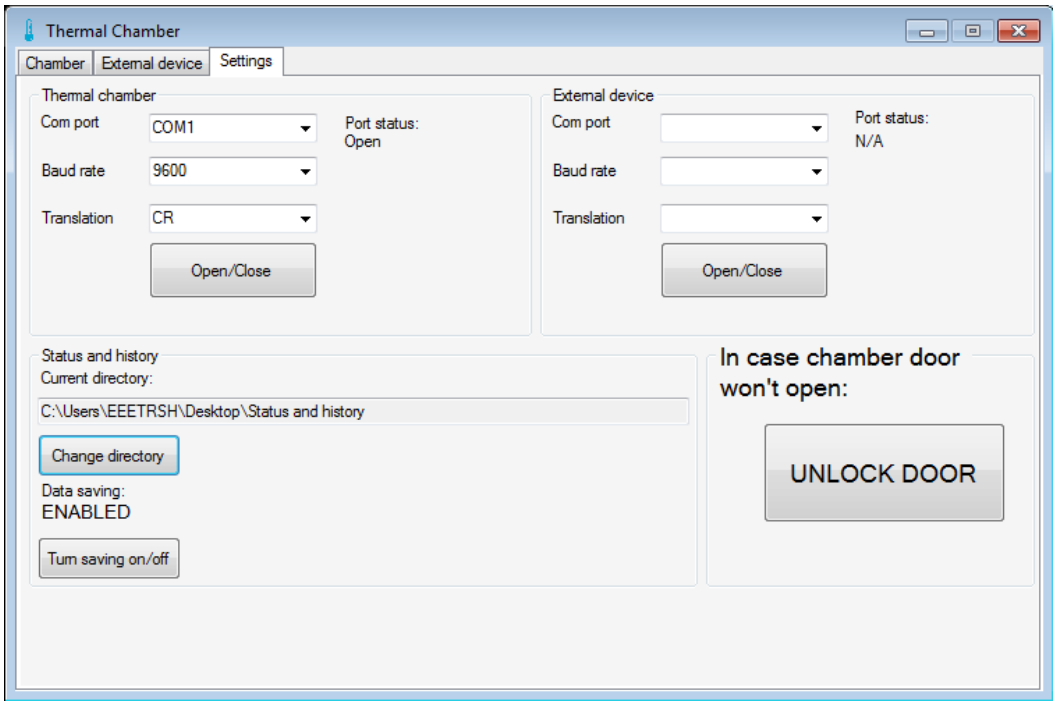
Kasutajaliidese poolt pakutav funktsionaalsus sai grupeeritud kolme rühma: temperatuurikambri töö ja mõõtmistega seotud, iseseisva seadmega suhtlemine ning sätted kambri ja iseseisva seadme töö juhtimiseks. Nende kolme rühma funktsionaalsus on jagatud kasutajaliidese akna kolme vahekaardi vahel ära, milleks on: *Chamber* (Joonis 5), *External device* (Joonis 6) ja *Settings* (Joonis 7).



Joonis 5. Kasutajaliidese temperatuurikambri juhtimise vahekaart.



Joonis 6. Kasutajaliidese iseseisva seadme suhtluse vahekaart.



Joonis 7. Kasutajaliidese sätete vahekaart.

4.1 Kambri hetke temperatuuri kuvamine

Temperatuurikambri hetketemperatuuri kuvatakse vahekaardil *Chamber* (ee: Kamber), koondkastis *Current temperature* (ee: Hetke temperatuur) (Joonis 5. Kasutajaliidese temperatuurikambri juhtimise vahekaart.Joonis 5).

Temperatuurikambri hetketemperatuur uuendamine on kasutajaliideses teostatud taimeriga, mis iga 1 sekundilise intervalli järel saadab kambri kontrollerile käsu „TEMP?“ (Tabel 1) läbi jadapordi. Ainuke erand on üheksanda intervalli möödumisel, millal küsitakse hoopis jahutusvedeliku temperatuuri. Temperatuurikamber vastab saadetud käsule kambri hetke temperatuuriga formaadis „+/- xx.x“. Kui kasutajaliides saab vastuse kätte, kuvatakse see koondkastis *Current temperature*.

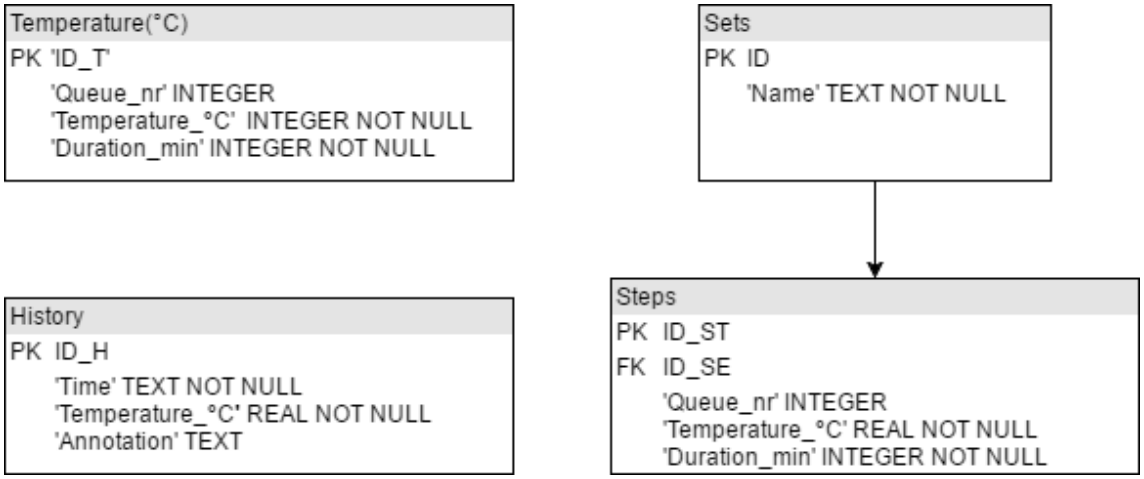
Eraldi intervall on jahutusvedeliku temperatuurile eraldatud, et eristada jahutusvedeliku temperatuur kambri temperatuurist, kuna temperatuurikamber vastab mõlemale päringule samas formaadis ja seega on need teineteisest eristamatud. Sellise eraldamisega välditakse olukorda, kus programm kasutab jahutusvedeliku temperatuuri kambri temperatuurina ja vastupidi.

Ühesekundiline intervall sai valitud, kuna kamber on võimeline muutma temperatuuri 0.2 kuni 0.5 kraadi Celsiuse võrra sekundis. Valitud intervall võimaldab kasutajal määrata ära kas temperatuurikamber töötab korrektselt ehk kas temperatuur tõuseb või langeb vastavalt sellele antud käsule. Mõningatel juhtudel võib eelmisest temperatuurist kõrgema määramine põhjustada kambris hetkelist temperatuuri langust. See tuleneb kompressoris olevast külma õhu säilimisest, mis puhutakse kambrisse. Kambri ettenähtud töötsükkel normaliseerub peale paarikümmet sekundit. Pikema aja möödudes on kasutajale selge, et süsteem on vigane ja pole seega töökorras.

4.2 Andmete hoiustamine

Temperatuurikambrilt saadud andmed säilitatakse SQLite kohalikus andmebaasi failis database.db3. SQLite sai valitud, kuna see on lihtsa ülesehitusega ning ei vaja andmete hoiustamiseks võrgus olevat serverit. SQLite andmebaasi maksimaalne maht on 140 terabaiti. Pikim sisestatav string võib olla kuni miljard baiti suur ning teoreetiliselt mahub tabelisse 2^{64} rida.

Eelnimetatud andmebaas koosneb neljast andmebaasitabelist: *Temperature(°C), Steps, Sets* ja *History* (Joonis 8). Andmebaas luuakse programmi käivitamisel, kui seda eelnevalt ei eksisteeri programmi lähtekaustas. Tabelitest *Temperature(°C), Steps* ja *Sets* on võimalik sissekandeid kustutada nuppudega *Delete selected step/set* ja *Delete all steps/sets*. Tabel *History* tühjendatakse iga kord kui programm käivitatakse, kuid nupuga *Clear graph* on võimalik seda teha ka manuaalselt. Tabeli tühjendamine toimub programmi käivitamisel, et vältida iganenud andmete jäämist andmebaasi, kui programm peaks mingil põhjusel ootamatult sulguma.



Joonis 8. Andmebaasi tabelid.

Andmebaasitabel *Temperature(°C)* hoiab endas kasutaja poolt loodud käske, mis lähevad temperatuurikambrile käitamiseks. Tabel koosneb neljast tulbast: ID_T, Queue_nr, Temperature_°C, Duration_min. Esimene tulp, ID_T, sisaldab unikaalseid identimisnumbreid. Teine tulp, Queue_nr, on järjekorranumbrite jaoks. Kolmandas tulbas, Temperature_°C, on täisarvulised temperatuuride väärtused. Viimane tulp, Duration_min, hoiab endas täisarvulisi aegu minutites.

Andmebaasitabeleid *Sets* ja *Steps* kasutatakse kasutajate poolt loodud käsukomplektide hoidmiseks. Tabel *Sets* koosneb kahest tulbast: ID ja *Name*. Esimene tulp, ID, on unikaalsete identimisnumbrite jaoks. Teine tulp, *Name*, on mõeldud hoidma kasutajate poolt sisestatavaid nimesid komplektidele.

Andmebaasitabel *Steps* hoiab komplekteerituid käske, mis on seotud andmebaasitabeli *Sets* sissekannetega. Tabel *Steps* koosneb viiest tulbast: ID_ST, Queue_nr, Temperature_°C, Duration_min ning ID_SE. Esimesed neli tulpa täidavad samu

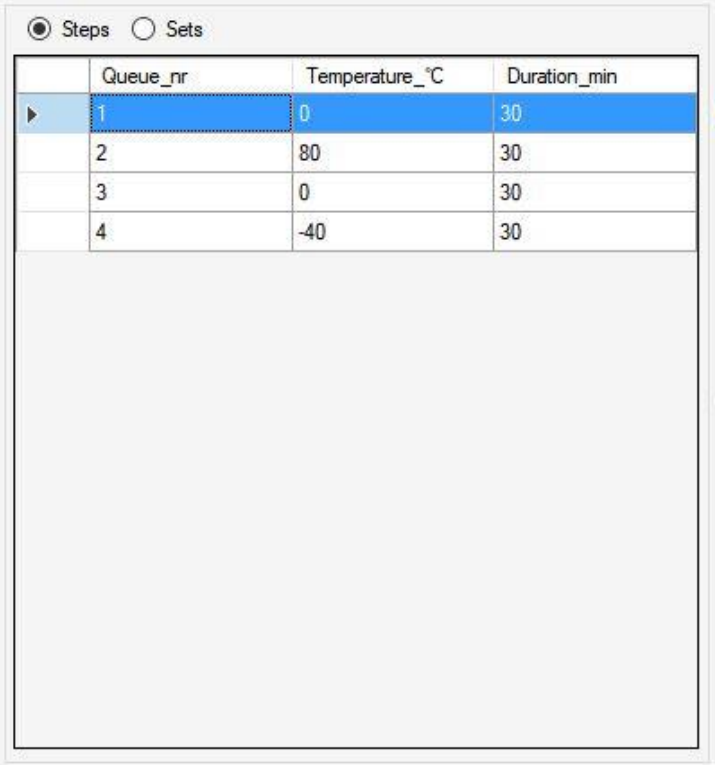
ülesandeid nagu tabeli *Temperature(°C)* tulbad. Viimane tulp, ID_SE, sisaldab viitenumbreid tabeli *Sets* identimisnumbritele ID tulbas.

History andmebaasitabelis hoitakse temperatuurikambrilt saaduid andmeid, kui temperatuurikamber täidab käske. Tabel koosneb neljast tulbast: ID_H, *Time*, *Temperature_°C* ja *Annotation*. Esimene tulp, ID_H, on identimisnumbrite jaoks. Teine tulp, *Time*, on mõeldud sissekande aja salvestamiseks. Tulbas *Temperature_°C* hoitakse temperatuure ning viimases tulbas, *Annotation*, hoitakse käske tekstina.

Kuna iga telekommunikatsiooniseadme testimine on erinev, vastavalt testi väljatöötajatele, siis on raske hinnata, kui palju sissekandeid iga testiga andmebaasi tehakse. Temperatuurikambris teostatavad testid võivad kesta mitu kuud, kuid ettevõtte soovis esialgu, et teste oleks võimalik teostada kuni 48 tundi. Ettevõtte poolt läbiviidud esialgne test koosnes kõigest neljast sissekandest andmebaasitabelisse *Temperature(°C)* ning kestis ligikaudu kolm tundi. Kõige rohkem sissekandeid tehti andmebaasitabelisse *History*. Sissekannete arv *History* tabelis oli ligikaudu 1080, kui võtta testi pikkuseks täpselt 3 tundi ehk 10800 sekundit ja teades, et testi ajal tehakse sissekanne andmetabelisse *History* iga 10 sekundi möödumisel (peatükk 4.5). Autor ise viis läbi katse, kus *History* andmebaasitabelisse tehti sissekandeid kuueteist tunni vältel, mis tegi kokku ligikaudu 5760 sissekannet. Mõlemal korral toimis programm normaalselt ja polnud põhjust arvata vastupidist ka rohkemate sissekannetega. Seetõttu peeti SQLite baasmootorit piisavaks.

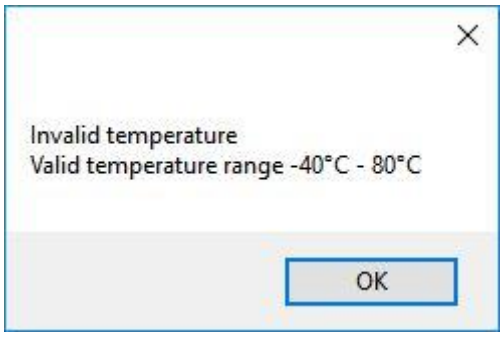
4.3 Käskude loomine, kustutamine, kuvamine ning saatmine temperatuurikambrile käitamiseks

Kasutajaliides võimaldab kasutajal luua käsujada (Joonis 9. Näide käskudest käsujadas.Joonis 9) ning saata selle temperatuurikambri kontrollerile käitamiseks. Käsujada koosneb üksikutest käskudest, mida käitatakse nende sisestamise järjekorras. Käske nimetatakse sammudeks (en: *Step*). Samuti on kasutajal võimalik salvestada loodud käsujada koondnimega ning hiljem salvestatud käsujadad saata käitamiseks temperatuurikambrile. Koondnimega salvestatud käske nimetatakse komplektideks (en: *Sets*).

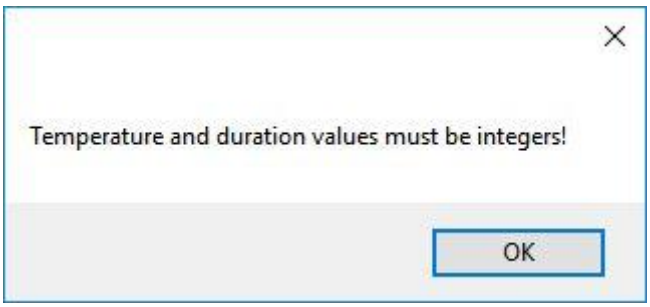


Joonis 9. Näide käskudest käsujadas.

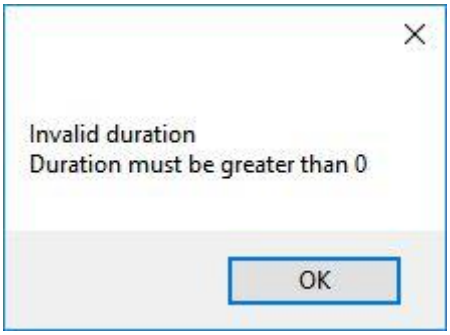
Käsk koosneb kasutaja poolt määratud temperatuurist Celsiuse skaalal ning kestvusest minutites (Joonis 9). Käsud sisestatakse vahekaardilt *Chamber* (Joonis 1Joonis 5) sisestades soovitud temperatuuri tekstikasti *Temperature(°C)* (ee: Temperatuur(°C)) ja selle hoidmise kestvuse minutites tekstikasti *Duration(min)* (ee: Kestvus(min)) ning seejärel vajutades nuppu *Add step* (ee: Lisa samm). Nupule on võimalik vajutada ainult siis, kui mõlemas tekstikastis on mingisugune väärtus, et vältida tühja käsu loomist. Nupule vajutusega luuakse uus sissekanne SQLite andmebaasitabelisse *Temperature(°C)*. Temperatuuri sisestus on piiratud vahemikus −40°C kuni 80°C, mis vastab temperatuurikambri äärmustemperatuuride, mille vastu eksimisel sissekannet ei tehta ning programm kuvab akna tekstiga „*Invalid Temperature: Valid temperature range -40-80°C*“ (ee: „Kehtetu temperatuur: Kehtiv temperatuurivahemik -40-80°C“) (Joonis 10). Sisestatav temperatuur ja kestvus peavad olema täisarvud. Murdarvude sisestamisel kuvatakse aken tekstiga: „*Temperature and duration values must be integers*“ (ee: „Temperatuuri ja kestvuse väärtused peavad olema täisarvud“) (Joonis 11). Lisaks peab sisestatav kestvus olema ka suurem nullist. Kui sisestatud kestvus on null või sellest väiksem kuvatakse aken tekstiga: *„Duration must be greater than 0“* (ee: „Kestvus peab olema nullist suurem“) (Joonis 12).



Joonis 10. Vales vahemikus sisestatud temperatuuri teavitusaken.

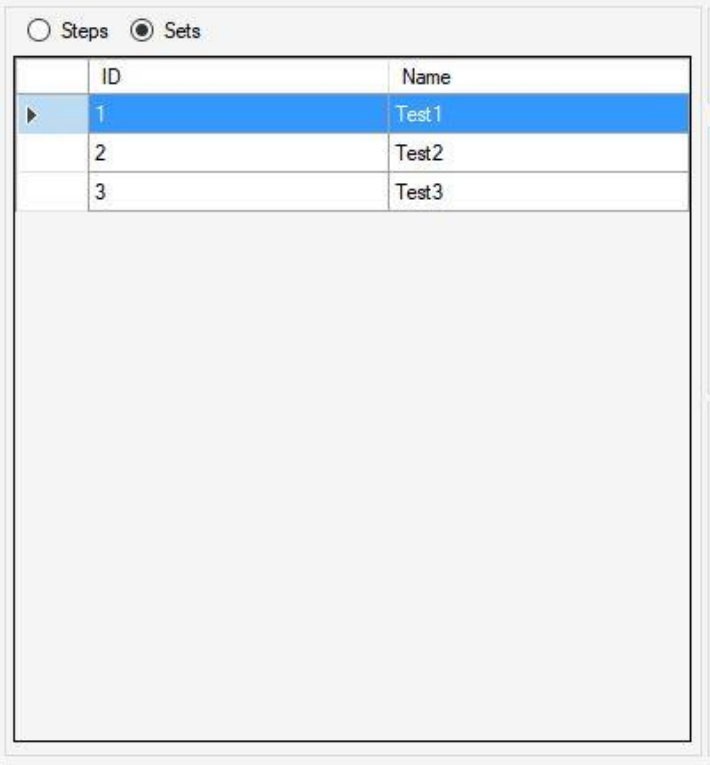


Joonis 11. Murdarvude sisestamise teavitusaken.

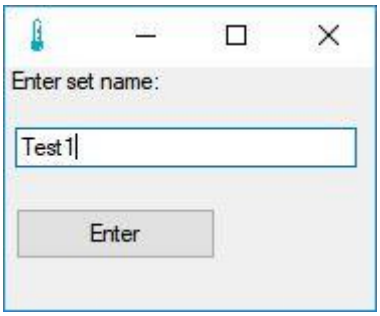


Joonis 12. Negatiivse või null kestvuse teavitusaken.

Käskude salvestamine komplektiks (Joonis 13) toimub nupule *Save set* (ee: Salvesta komplekt) vajutamisega, peale mida avatakse aken tekstikastiga *Enter set name* (ee: Sisestage komplekti nimi) (Joonis 14). Kui kasutaja on sisestanud komplektile soovitud nime tekstikasti ja vajutanud nupule *Enter* (ee: Sisesta), siis programm loob andmebaasitabelisse *Sets* (ee: Komplektid) vastava nimega sissekande ning annab sellele unikaalse identimisnumbri. Seejärel kopeerib programm kõik andmebaasitabelis *Temperature(°C)* olevad käsud andmebaasitabelisse *Steps* (ee: Sammud) ning annab neile viitenumbri, mis viitab tabelis *Sets* vastavale komplekti identimisnumbrile (Joonis 8).



Joonis 13. Salvestatud komplektid.



Joonis 14. Komplektide salvestamise aken.

Loodud käsud ning komplektid kuvatakse vahekaardil *Chamber* tabelis, mis võetakse vastavalt SQLite andmebaasitabelitest *Temperature(°C)* ja *Steps*. Raadionuppudega *Steps* ja *Sets* on võimalik valida käsujada ning komplektide kuvamise vahel vahekaardil. Raadionupu *Steps* valimisel, mis on ka käivitamisel vaikimisi valitud, kuvatakse vahekaardil käsujada, mis asub andmebaasitabelis *Temperature(°C)* (Joonis 9). Käsud kuvatakse järjekorranumbri, temperatuuri ning kestvusega. Raadionupu *Sets* valimisel kuvatakse andmebaasitabelis *Sets* olevad andmed vahekaardil (Joonis 13). Komplektid kuvatakse nende unikaalse identifikatsiooni numbriga ning nimega.

Vahekaardil olevaid käske ja komplekte on võimalik vasaku hiireklahvi vajutusega ükshaaval esile tõsta, mis kuvatakse tabelis siniselt (Joonis 5). Esiletõstetud käske ja komplekte on võimalik vastavalt märgistatud raadionupule kustutada nupuga *Delete*

selected step/set (ee: Kustuta valitud samm/komplekt), mis kustutab SQLite andmebaasi tabelist *Temperature(°C)*või *Sets* vastava käsu või komplekti ning kuvab uuendatud tabeli vahekaardil *Chamber*. Nupuga *Delete ALL steps/sets* (ee: Kustuta KÕIK sammud/komplektid) on võimalik kustutada kõik tabelis olevad käsud või komplektid.

Vahekaardil *Chamber* tabelis esile tõstetud komplekti käske (Joonis 9) on võimalik kuvada, kui kasutaja vajutab nuppu *Load set* (ee: Lae komplekt), mille peale programm kopeerib andmebaasitabelis *Steps* kõik sissekanded, mis viitavad esile tõstetud komplektile, ning asendab tabelis *Temperature(°C)*olevad sissekanded nendega. Seejärel märgistab programm raadionupu *Steps* ning kuvab vahekaardil andmebaasitabeli *Temperature(°C)* käsud.

Nupuga *Start* on võimalik saata andmebaasitabelis *Temperature(°C)*olevad käsud temperatuurikambri kontrollerile käitamiseks vastavalt nende järjekorranumbrile. Viimati käitatud käsk kuvatakse sildi *Current step:* all (Joonis 5). Järelejäänud aeg järgmise käsu käitamiseni kuvatakse jooksvalt sekundi täpsusega sildi *Time remaining* all. Kui temperatuurikamber pole saavutanud käsuga määratud temperatuuri +/− 1 kraadise täpsusega, kuvatakse sildil all tekst N/A ehk aeg pole veel saadaval.

Programm ei kuva terve käsujada läbiviimiseks kuluvat aega, kuna temperatuuri tõusu ja langust on väga raske ette ennustada. Temperatuuri muutumise kiirus on tingitud kambri hetke temperatuuri ning küttekeha (temperatuuri tõstmisel) või jahutusvedeliku (temperatuuri langetamisel) temperatuuri vahest. Mida suurem on kambri ja küttekeha/jahutusvedeliku temperatuuride vahe, seda äkilisemalt toimub temperatuurimuutus. Etteennustamise teeb raskemaks veel asjaolu, et jahutusvedeliku temperatuur võib töötsükli alguses ja lõpus olla erinev.

Stop nupp, mis asub *Start* nupu all (Joonis 5), võimaldab peatada edasise käskude saatmise temperatuurikambri kontrollerile. Nupule vajutus tühistab käskude edastamisjärjekorra, peale mida nupule *Start* vajutusel alustatakse käskude saatmist kontrollerile esimesest sisestatud käsust.

4.4 Temperatuurikambri töövalmidus

Temperatuurikambri temperatuuride hoidmine sõltub väga palju jahutusvedeliku temperatuurist. Mida madalam on jahutusvedeliku temperatuur, seda täpsemalt

suudetakse soovitud temperatuuri hoida. Seetõttu on kasutajaliidesele lisatud indikaator, mis näitab kas süsteem on töövalmis või mitte.

Temperatuurikambri töötsükli jälgimise tulemusena erinevate jahutusvedelike temperatuuride juures selgus, et jahutusvedelik peab olema kõige rohkem −20 kraadi Celsiust. Sellest kõrgematel jahutusvedelike temperatuuridel on temperatuuri kõikumine kambris rohkem kui 5 kraadi Celsiust, mida meeskond pidas mitte kõlbulikuks temperatuurivahemikuks telekommunikatsiooniseadmete testimiseks. Seadmete testimisel on oluline, et temperatuur ei kõiguks palju, kuna see vähendab läbiviidud testide usaldusväärsust. Mida suurem on temperatuuri kõikumine, seda vähem viibib testitav seade ettenähtud temperatuuril. Samuti võivad liiga suured kõikumised temperatuuris kahjustada materjale, millest seadmed koosnevad.

Töövalmiduse indikaator koosneb tekstiväljast ja pildist, mis asub *Chamber* vahekaardil sildi *Test stability* all. Vastavalt jahutusvedeliku temperatuurile kuvatakse tekstiväljal *Stable* või *Unstable* ja neile vastavalt pilt rohelisest või punasest ringist. Tekst *Stable* ja pilt rohelisest ringist kuvatakse juhul kui jahutusvedeliku temperatuuri on alla −20 kraadi Celsiust. Kui temperatuuri on kõrgem kui −20°C, siis kuvatakse tekst *Unstable* ja pilt punasest ringist (Joonis 5).

Jahutusvedeliku temperatuuri kontrollitakse käsuga „TEMP:CL?“. Käsk saadetakse sama taimeri 9 intervalli möödumisel, mis on ligikaudu 9 sekundit. Ajaliselt sai valitud 9 intervalli seetõttu, et kui jahutusvedelik on saavutanud madala temperatuuri, siis temperatuurikambri töötamise ajal see enam ei lange ja seega pole vajalik seda iga sekund kontrollida. Ettevaatusabinõuna jätkatakse kontrollimist ka pärast -20°C saavutamist, et jälgida kas temperatuurikambri jahutussüsteem on töökorras.

4.5 Andmete kuvamine graafikul

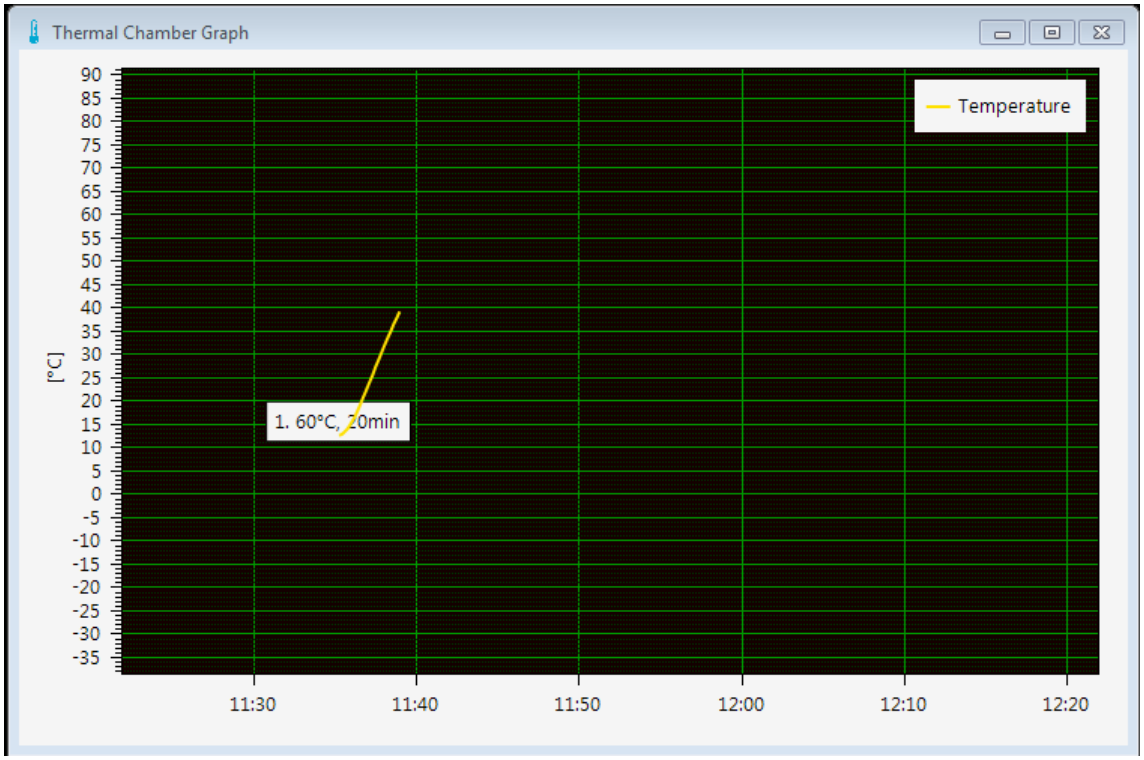
Arendatud temperatuurikambri kasutajaliides võimaldab kasutajal kuvada kambri tööprotsessi graafikul programmi siseselt alates esimesest käsujada käitamise hetkest. Graafik kuvatakse eraldi aknas, mida on võimalik avada vahekaardilt *Chamber* koondkastist *Graph* nupuga *Show graph.* Graafiku loomiseks on kasutatud OxyPlot teeki.

Graafik kuvab temperatuuri, aega ning käsu käitamist temperatuuri ja aja telgedel. Käsu käitamine kuvatakse graafikul annotatsioonina vastaval temperatuuril ja ajal, millal see

edastati temperatuurikambri kontrollerile. Graafiku kõiki sissekandeid on võimalik kustutada nupuga *Clear graph*.

Graafikut uuendatakse iga 10 sekundi järelt, mis arvestatakse taimeri 10 intervalli möödumisel, mida kasutatakse temperatuurikambri hetke temperatuuride küsimiseks kontrollerilt. Samal ajal kirjutatakse ka graafiku andmed andmebaasitabelisse *History*. Kümne sekundiline periood sai valitud graafiku uuendamiseks, kuna graafik on mõeldud kogu tööprotsessi jälgimiseks tagantjärele. Telekommunikatsiooniseadmete testimisel pole kriitiline teada täpset temperatuuri, mille all seade lühikesel ajahetkel (näiteks sekundil) oli, vaid kui kaua seade oli nõutud keskmisel temperatuuril.

Disain graafikule sai valitud vastavalt sellele, mis tundus autorile ja meeskonna liikmetele kõige meeldivam. Seetõttu sai valitud must taust ning graafiku punkte ühendab kollane joon. Joonestik on tumeroheline ja jooned, mis tähistavad telgedel kuvatuid väärtusi, on heledamad, kui jooned, mis tähistavad kuvamata väärtusi (Joonis 15).



Joonis 15. Temperatuurikambri temperatuuride graafik.

Graafikule hakatakse andmeid kuvama, kui *Start* nuppu on vajutatud. Andmete kuvamine katkestatakse *Stop* nupule vajutamisega.

Graafikul navigeerimiseks on jäetud Oxyplot teegi vaikesätted. Vastavalt vaikesätetele on graafikul võimalik panoreerida, telgedel kuvatavaid vahemikke suurendada või vähendada.

Graafiku panoreerimine toimub arvutihiire parema klahvi all hoidmisel ning hiire graafikult üle lohistamisel. Sama on võimalik teha ka arvutiklaviatuuril nupu „Alt“ ja arvutihiire vasaku klahvi all hoidmisel ning seejärel hiire graafikult üle lohistamisel.

Telgedel kuvatavate vahemikkude suurendamine või vähendamine graafikul toimub arvutiklaviatuuril nupu „Ctrl“ all hoidmisega ning arvutihiire rulliku kerimisega tagasi või edasi. Vahemike suurendamine ja vähendamine oleneb hiire asetusest graafikul: hoides hiirt graafiku võrgustikul valitakse lähtepunktiks, mille suhtes vahemikke suurendatakse või vähendatakse, hiire positsioon. Hoides hiirt telje peal on võimalik ühte telge korraga muuta ilma teist muutmata. Kasutajal on võimalik valida ka kindel nelinurkne ala, mida soovitakse suurendada. Selleks tuleb hoida all klaviatuuril klahvi „Ctrl“ ja arvutihiirel paremat klahvi või rullikut ja libistada hiir diagonaalselt üle graafikul soovitud ala. Sama on võimalik teha ka veel hiire vasaku klahviga, toimides eelnevalt kirjeldatud korrale ja hoides lisaks all veel klaviatuuriklahvi „Alt“.

4.6 Graafiku eksportimine Microsoft Excel formaati

Graafikul olevad andmed asuvad *History* andmebaasitabelis ja nende põhjal on võimalik kasutajal luua Microsoft Exceli .xls fail, mis sisaldab kõiki neid andmeid ja kuvab need graafikul. Exceli .xls formaat sai valitud seetõttu, et see käivitub kõikidel Microsoft Excel versioonidel.

Andmete eksportimine algab *Chamber* vahekaardil *Graph* koondkastil nupule *Export as .xls* vajutamisega (Joonis 5), kui *History* andmebaasitabelis on sissekandeid, nende puudumisel kuvatakse aken tekstiga „*No values to export!*“ (ee: „Pole väärtuseid, mida eksportida!“) (Joonis 16). Eksportimisel on tegemist aeganõudva protsessiga, mis võib kesta mõnest sekundist kümnete minutiteni. Protsessi pikkus oleneb eksporditavate andmete mahust. Seetõttu temperatuuri pärimise taimeri viidet pikendatakse lõpmatuseni ning peale eksportimise lõppu seatakse tagasi 500 ms peale. Peale viite pikendamist kontrollib programm kas arvutis on Excel olemas. Exceli puudumisel kuvatakse kasutajale aken tekstiga: „*Microsoft Excel not found*“ (ee: „Microsoft Excelit ei

leitud“)ning eksportimine katkestatakse. Kui programm on olemas, siis luuakse Exceli tabel 4 tulbaga, mille ridade arv on võrdne sissekannetega *History* andmebaasis. Loodud 4 tulpa on *Time*, *Temperature*, *Step* ja neljas tulp on nimetu. Esimesse tulpa, *Time*, kirjutatakse andmebaasitabeli tulba *Time* kõik väärtused. *Step* tulpa kirjutatakse andmebaasitabeli tulba *Annotation* väärtused. Nimetusse tulpa kirjutatakse andmebaasi tulbas *Temperature_°C* olevad väärtused. Nimetu tulp on kohahoidja temperatuuride tekstiväärtustele, mis loetakse arvuliste väärtustena tulpa *Temperature*. Selline dubleering on vajalik, kuna andmebaasitabelist saadud temperatuuride väärtused on tekstivormingus ja sisaldavad murdarvudena punkti mitte koma, mille tõttu Excel muudab need kuupäevadeks, kui neid lugeda graafikusse.

Joonis 16. Väärtuste ekspordiks puudumise teavitusaken

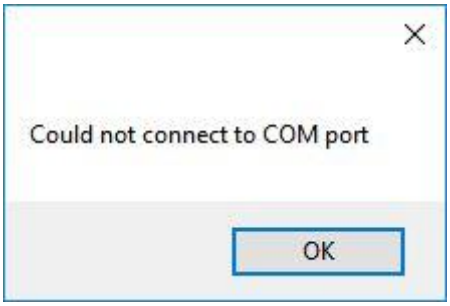
Kui programm on lugenud kõik andmed Exceli formaati avatakse kasutajale .xls faili salvestamisaken. Peale asukohakataloogi valimist luuakse .xls fail sinna kasutaja poolt valitud nimega. Nupule *Cancel* vajutamisega faili ei looda.

4.7 Kasutajaliidese sätted tööks

Temperatuurikambri juhtimise kasutajaliides võimaldab kasutajal valida jadapordi, mille kaudu suheldakse temperatuurikambri kontrolleriga. Samuti on võimalik valida boodi kiirus ning tõlgendus suhtlusele. Iseseisva seadmega suhtluse loomiseks on olemas samad valikud.

Temperatuurikambriga ja arvuti vahelise ühenduse loomiseks on seatud vaikimisi jadaport COM1, mille boodi kiiruseks on 9600, puudub paarsusebit ning tõlgendus on seatud reavahetuse peale. Programm püüab igal käivitamisel ühendada arvuti temperatuurikambriga lähtudes eelnimetatud sätetest, seda põhjusel, et tööks mõeldud arvuti, mille jaoks saigi antud kasutajaliides arendatud, kasutab suhtluseks just neid seadeid. Kui käivitamisel ei õnnestu mingil põhjusel luua ühendust

temperatuurikambriga, olgu selleks siis jadapordi COM1 hetkeline kasutusel olek mõne teise programmi poolt või pordi COM1 puudumine masinast, siis annab programm sellest märku kutsudes välja teavitusakna tekstiga: „Could not connect to COM port“ (ee: „Ei olnud võimalik luua ühendust jadapordiga“) (Joonis 17). Juhul kui ühendus COM1-ga on olemas, kuid saadetud päringutele vastuseid ei tule 500 ms möödumisel, näiteks kaabliühenduse puudumise tõttu temperatuurikambriga, siis suletakse ühendus jadapordiga ning programm kutsub välja akna tekstiga: „*Thermal chamber not connected*“ (ee: Ei olnud võimalik saata andmeid temperatuurikambrile).



Joonis 17. COM1 ühenduse loomise nurjumise teavitusaken.

Juhul kui programmi käivitamisel ei õnnestunud kasutajaliidesel temperatuurikambriga mingil põhjusel ühendust luua, siis on kasutajal võimalik peale teavitusakna sulgemist sätete vahekaardilt (en: *Settings*) valida *Thermal chamber* (ee: Temperatuurikamber) koondkastis saadaval olevad sätted, mis võivad aidata ühenduse luua manuaalselt (Joonis 7).

Koondkast *Thermal chamber* sisaldab endas kolme liitkasti: *COM port* (ee: Jadaport), *Baud rate* (ee: boodi kiirus) ja *Translation* (ee: Tõlgendus). Liitkast *COM port* võimaldab kasutajal valida arvutis saadaval olevate jadaportide vahel. Liitkast *Baud rate* võimaldab kasutajal valida erinevate boodi kiiruste vahel. Liitkast *Translation* võimaldab kasutajal valida 3 eri tõlgenduse vahel: CR, LF ja CR+LF. Ühenduse loomiseks on koondkastis nupp „*Open/Close*“ (ee: Ava/Sulge), mis võimaldab kasutajal luua ühendus temperatuurikambriga või see sulgeda. Ühenduse olemasolu või selle puudumine kuvatakse sildi all „*Port status*“ (ee: „Jadapordi olek“), millel on kaks eri väärtust: „*Open*“ või „*Closed*“ (ee: „Avatud“ või „Suletud“). Ühenduse olek kuvatakse lisaks veel vahekaardil *Chamber* koondkastis *Chamber port* (ee: Kambri port) sildi *Status* (ee: Olek) all.

Ühenduse loomine iseseisva seadmega on analoogne temperatuurikambriga ühenduse loomisega. Iseseisva seadmega ühenduse loomine toimub samuti vahekaardil *Settings,* koondkastis *External device* (ee: Väline seade). Koondkast sisaldab samu liitkaste, nuppe ning silte, mida sisaldab ka koondkast *Thermal chamber*.

4.8 Iseseisva seadmega suhtlemine

Temperatuurikambri arvuti külge saab jadaportide kaudu ühendada veel seadmeid ja seega on programmi lisatud veel lihtne terminal sellise seadmega suhtlemiseks. Terminal on lisatud kasutusmugavuse pärast, et vältida mitmete programmi akende lahtiolekute vajadust. Iseseisva seadme all on mõeldud seadet, mis on temperatuurikambrist eraldiseisev üksus ega mõjuta kambri enda tööd. Iseseisva seadme suhtlus on mõeldud küll DUT jaoks, kuid seade ei pea kambris olema, et sellega suhelda.

Iseseisva seadmega on võimalik suhelda läbi kasutaja poolt valitud jadapordi ja seadistustega *External device* vahekaardilt (Joonis 6). Jadapordi ja selle sätted saab panna paika sätete vahekaardilt *External device* koondkastist analoogiliselt temperatuurikambri enda seadete sätestamisele (Joonis 7).

External device vahekaardil on kuvatud jadapordi olek: *Opened* või *Closed*. Selle all asub tekstikast, milles kuvatakse saadetud käsud ja vastuvõetud vastused. Saadetud käsud kuvatakse formaadis: „CMD: [käsk] [kellaaeg]“. Vastuvõetud vastused kuvatakse formaadis: RESPONSE: [vastus] [kellaaeg]. Kellaaeg kuvatakse 24 tunni formaadis ja vastavalt arvuti enda kellale.

Käske on võimalik saata nupule *Send* vajutamisega. Eelnimetatud nupule vajutamisega saadetakse tekstikastis, mis asub vastuseid ja käske kuvava tekstikasti all, olev tekst jadapordi saatmispuhvrisse. Kasutatavale jadapordile pole pandud viite taimerile piirangut, kuna kasutaja kirjavea tõttu ei pruugi programm vastust saada.

Kasutajal on võimalus automaatselt saata tekstikastis olevat teksti enda soovitud aja tagant. Seda on võimalik teha vajutades nupule *Start/Stop loop*. Aeg, mille tagant hakatakse teksti saatma, määratakse ära kahe numbrikastiga, millest üks võimaldab seada minuteid ja teine sekundeid. Sildi *Loop Count* all kuvatakse saatmiste arv ja sildi *Current loop time* all kuvatakse järelejäänud aeg järgmise saatmiseni.

5 Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli arendada tarkvaraline rakendus Ericsson Eesti Tallinna tehasele, mis võimaldaks kasutajal juhtida modifitseeritud temperatuurikambri tööd, koguda ning töödelda sellelt saadavaid andmeid ning võimaldaks suhelda ühe iseseisva seadmega. Nimetatuid kriteeriume pidi rakendus täitma graafilise kasutajaliidesena, mis töötab operatsioonisüsteemi *Microsoft Windows* keskkonnas.

Kasutajaliidese arendamiseks oli autoril vajalik tutvuda temperatuurikambriga ning sellega suhtlemisega. Selgus, et temperatuurikambriga suhtlemine toimub läbi arvuti ja temperatuurikambri kontrolleri RS232 jadaportide vastavalt asünkroonsele järjestikedastusele. Kui arvutis saata kindlaid sõnumeid kontrollerile on võimalik temperatuurikambri tööd juhtida ning kontrollida.

Tarkvara arendamisel valis autor võimalikud lihtsad lahendused temperatuurikambri kasutajaliidese ja suhtluse arhitektuurile, et vastata Ericsson Eesti Tallinna tehase tootetehnoloogia meeskonna soovile pakkuda funktsionaalset tarkvara võimalikult lühikese ajaga. Autor valis tarkvara arendamiseks klient-server arhitektuuri, kus arvuti täitis nii kliendi kui ka serveri ülesandeid. Tarkvara funktsionaalsuse loogiline eraldamine kirjeldati kolmekihilise arhitektuurina. Autor jättis tarkvara arhitektuuri kihid üksteisest sõltuvaks arhitektuuri rakendamise lihtsuse tõttu.

Tarkvara arendati vastavalt autori poolt koostatud ja Ericssoni pool kinnitatud funktsionaalsetele ja mitte-funktsionaalsetele nõuetele. Nõuded said paika pandud vastavalt Ericsson Eesti Tallinna tehase tootetehnoloogia meeskonna kirjeldustele, mida süsteemilt oodati.

Arendusprotsessi lõpus suutis temperatuurikambri juhtimise kasutajaliides pakkuda kasutajale sellelt soovitud funktsionaalsust.

Kasutatud kirjandus

[1] Ericsson AS, „Temperature Chamber Control SW,“ 2009.

[2] TAL Technologies, „Introduction to Serial Communications“.

[3] Microsoft Corporation, „Chapter 5: Layered Application Guidelines,“ %1 *Microsoft Application Architecture Guide*, teine trükk toim., Microsoft Corporation, 2009.

[4] M. Fowler, Patterns of Enterprise Application Architecture, Boston: Addison-Wesley, 2002.