

Projekt **SIMULATOR** za upravljanje prostora

Naročnik: PROstor d. o. o.
Vodja projekta: Primož Volker

Status dokumentov

- [*] D1. Naročnikove zahteve
- [*] D2. Plan projekta
- [*] D3. Sistemske specifikacije
- [*] D4. Testni primeri
- [*] D5. Poročilo o preverjanju
- [*] D6. Načrtovalska dokumentacija
- [*] D7. Uporabniški priročnik

Začetek: 15. 10. 2020

Konec: 7. 2. 2021

Ime in priimek	Vloga	Naslov	Opomba
Deni Vinšek	Preverjanje	deni.vinsek@student.um.si	
Primož Volker	Razvoj	primoz.volker@student.um.si	

Rubrika, ki je namenjena pregledovalcu projekta

Projekt oddan:

D1 NAROČNIKOVE ZAHTEVE

Dokument	Naročnikove zahteve, Verzija 2.0, 22. 10. 2020
Naročnik	Podjetje PROstor d. o. o.
Lokacija dokumenta	datoteka nar_zahteveOPI_2020_v2.pdf
Odgovorna oseba	Primož Volker

V podjetju PROstor d.o.o. se ukvarjamo z razvojem inovativnih rešitev na področju avtomatizacije in digitalizacije upravljanja poslovnih prostorov. Pri načrtovanju naših rešitev dajemo velik poudarek na okoljsko trajnost, energetsko učinkovitost ter ergonomičnost produktov, saj se zavedamo, da omenjene lastnosti pozitivno vplivajo tako na izboljšano uporabniško izkušnjo kot na optimizacijo poslovanja skozi nižanje stroškov. V podjetju smo prepoznali pomanjkanje rešitev, ki bi celovito naslovile problem zastarelosti poslovnih prostorov. V ta namen načrtujemo razvoj **centralnega sistema za upravljanje poslovnega prostora**, s čimer se nadejamo preboja na trg in s tem izboljšanja poslovnega uspeha. Skladno s stanjem na trgu optimizacije poslovnih prostorov smo v podjetju prilagodili naše storitve tako, da naslavljaajo probleme v povezavi s Covid-19 in za njih ponujajo visokotehnološke rešitve.

Projekt že ima izoblikovano idejno zasnovo, in sicer tako glede strojne opreme kot izgleda in funkcionalnosti. Sedaj smo v fazi iskanja resnega partnerja, ki bi prevzel razvoj programske opreme. Ker želimo preveriti osnovni koncept in delovanje centralnega sistema za upravljanje prostora, naj bo program napisan **v obliki simulatorja**. Najprej potrebujemo preprost simulator brez grafičnega vmesnika, ki bo izdelan kot konzolna aplikacija **v jeziku C++** v integriranem razvojnem okolju **Visual Studio**. Od simulatorja pričakujemo brezhibno in robustno delovanje v operacijskem sistemu **Windows**. Poleg tega mora biti simulator hiter in preprost za uporabo.

Simulator naj omogoča krmiljenje temperature, vlage in osvetljenosti prostora v osnovnem načinu delovanja. Predpogoj je, da uporabnik predhodno v tekstovno datoteko vpiše želene ambientalne lastnosti v obliki:

TEMPERATURA: vrednost [program zazna enoto samodejno, glede na vneseno vrednost.
Možne enote so °C, K in °F]
VLAZNOST: vrednost v obliki relativne vlažnosti [%]
OSVETLJENOST: 500 [lx]

V datoteki naj bo še:

INTERVAL TEMPERATURE: [10, 40] - enota so °C
STOPNJA VLAZNOSTI: [30,60] – relativna vlažnost v %
INTERVAL OSVETLJENOSTI: [10,10000] – enota so lx

Simulator naj pred pričetkom prebere vrednosti iz datoteke, nato pa naj omogoča izbiro med dvema režimoma delovanja – navadni in testni režim, pri čemer navadni režim obsega dva **načina delovanja**:

Testni režim:

Uporabnik v program vnese dejansko temperaturo v prostoru. Program naj omogoča prepoznavo merske enote vnesene temperature in njeno pretvorbo v ustrezno mersko enoto, ki se sklada z enoto želene ambientalne temperature. Nato naj izračuna razliko do želene temperature in izvede ukaz za regulacijo temperature. Analogno naj simulator omogoča vpis, izračun in izvedbo ukazov še za vlažnost in osvetljenost kot osnovni operaciji ter za prezračevanje in vsebnost razkužila kot napredni operaciji. Testni režim naj vsebuje še preostale preizkuse relevantne za testiranje navadnega režima delovanja simulatorja.

Navaden režim:

Avtomatski način: Računalnik naj si izmisli dejansko temperaturo na intervalu podanem v datoteki. Program naj v kolikor je to potrebno vrednost pretvori v mersko enoto, ki se sklada z enoto želene ambientalne temperature. Izmisli naj si še relativno stopnjo vlažnosti, in sicer med 30 in 60 %, ter osvetljenost na intervalu z datoteke. Nato naj za vsako posamezno meritev izračuna odstopanje od zelenih vrednosti ter izvede ukaze za popravek. Simulator naj uporabniku omogoča izbiro pri številu meritev in časovnem razmiku med njimi. Na koncu simulacije naj izračuna povprečno vrednost dejanskih ambientalnih vrednosti ter povprečno odstopanje od zelenih vrednosti za posamezne parametre.

Napredni (Covid-19 način): Napredni način omogoča enake funkcionalnosti, kot avtomatski način, pri čemer dodaja še funkcionalnost avtomatskega prezračevanja in spremljanja stanja razkužil. Računalnik naj si podobno kot pri avtomatskem načinu izmisli vsebnost CO₂ v zraku, in sicer na intervalu [200, 2500] ppm. Nato naj izračuna odstopanje do najbližje želene (optimalne) vrednosti ter izvede ukaz za popravek (tj. prehod v željen interval). Izračuna naj še povprečne vrednosti, analogno kot v avtomatskem načinu. Računalnik naj si izmisli še stanje razkužila, in sicer količino na intervalu od [1, 1000] ml. Na podlagi izmišljene količine naj o stanju razkužila informira uporabnika. Ker se količina razkužila ne spreminja tako hitro, kot ambientalne lastnosti, naj vsebnost razkužila preverja vsako 5 meritev, če je število meritev večje od 5 v nasprotnem primeru pa samo pri prvi iteraciji.

Izvajalec mora natančno **slediti internemu standardu, priporočilom za pisanje programske kode in poskrbeti za dokumentacijo**. Sestavni del projekta sta tudi **razvijalska dokumentacija in uporabniški priročnik**.

Od izvajalca pričakujemo, da do **29. 10. 2019** do **23.55** odda plan projekta, ki vključuje ceno. Program in dokumentacija morata biti oddana najkasneje **21. 1. 2021** do **23.55**. Projekt bo plačan po posameznih zaključenih fazah. Za vsak teden zamude bo odbitih 10 % plačila.

Maribor, 1. oktober 2020

direktor podjetja PROstor d.o.o.

D2 PLAN PROJEKTA

D2.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument: Plan Projekta	
verzija	1.0
datum	22. 10. 2020
lokacija	datoteka Projekt_Simulator_Vinsek_Volker.pdf
Odgovorni osebi	Primož Volker, Deni Vinšek
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve 2.0

D2.2 KRATEK OPIS PROBLEMA

Podjetje PROstor d.o. o. (v nadaljevanju naročnik) je 1. 10. 2020 predložila svoje zahteve, na podlagi katerih je bil izdelan ta plan projekta.

Naročnik se ukvarja z razvojem inovativnih rešitev na področju avtomatizacije in digitalizacije upravljanja poslovnih prostorov. V podjetju načrtujejo razvoj centralnega sistema za upravljanje poslovnega prostora, zato potrebujejo program v obliki simulatorja. Osnovna ideja projekta je že zasnovana, in sicer vsebuje zahteve strojne, kot tudi programske opreme in funkcionalnosti. Simulator mora v osnovi omogočati krmiljenje temperature, vlage in osvetljenosti prostora. Podpirati mora dva načina delovanja: testnega ter navadnega. Navaden se deli še na avtomatski in napredni (Covid-19) način. Podpirati mora tudi branje iz datoteke, kamor bodo vneseni vhodni podatki.

D2.2.1 Globalni cilji (globalne zahteve), ki jih želimo s produktom doseči

- 1) Program mora poskrbeti za pravilno simulacijo upravljanja posameznih sistemov v prostoru, saj se bo le tako lahko nadaljevala izvedba projekta v podjetju.
- 2) Program naj bo dovolj preprost za uporabo, da ne bo potrebno posebno usposabljanje operaterjev.

D2.2.2 Omejitve (operacijski sistem, aparturna oprema, standardi...)

- 1) Program mora teči na PC računalniku v operacijskem sistemu Windows kot konzolna aplikacija.
- 2) Za izvajanje programa Simulator mora zadoščati standardni PC. Program mora biti dokumentiran skladno s standardom CVVS 2/2000.

D2.2.3 Rok za zaključitev projekta, skupni stroški

- 1) 12. 2. 2021
- 2) Maksimalni skupni stroški izdelave projekta so 2000 EUR.

D2.2.4 Funkcije

Bistvene funkcije, ki jih mora sistem izvajati, da bodo doseženi globalni cilji:

- 1) Omogočati mora izbiro režima delovanja
- 2) Omogočati mora pridobivanje podatkov iz datotek
- 3) Poskrbeti mora za pravilno pretvorbo temperaturnih enot
- 4) Sistem mora vsebovati navodila za uporabo

D2.2.5 Pomembne karakteristike

- 1) Program se mora hitro (<1 s) odzvati na spremembe parametrov.
- 2) Program mora napačne ali nesmiselne vhodne podatke zavrniti in nadaljevati z delom, kot je najboljše možno.

D2.2.6 Neizvedljive zahteve

Zahteva za brezhibno delovanje bo zavrnjena, saj je popolna odsotnost napak nemogoča.

D2.2.7 Označevanje verzij

x.y DDMMLLLL

X glavna oznaka (velike spremembe), y za majhne spremembe, DDMMLLLL - datum

D2.3 ZAGOTAVLJANJE KAKOVOSTI (NAČRT PREVERJANJA)

D2.3.1 Objekti preverjanja

D1	Naročnikove zahteve
D2	Plan projekta
D3	Sistemske specifikacije
D4	Testne primere
D5	Poročilo o preverjanju
D6	Načrtovalsko dokumentacijo
D7	Uporabniški priročnik

Glede na izbran model razvoja obstajajo delni in končni produkti, ki jih je potrebno na koncu vsake faze preveriti (glej tabelo Pregled po produktih in aktivnostih). Kompletan terminski plan je podan v nadaljevanju tega dokumenta. Končni produkt predstavljajo dokumenti D1-D7.

A) Preverjanje programa v1.0

Program v1.0 bomo preverili s pregledom izvirne kode (stil kodiranja, skladnost s standardom) in testiranjem. Pripravljeni bodo določeni testni vzorci in postopki, ki jih bo natančneje definiral dokument Testni primeri. Preverjanje izvaja preverjevalec. Po preverjanju se izpolnijo pisna poročila o najdenih neustreznostih. Na podlagi teh poročil se izvede odpravljanje neustreznosti. Najprej se bodo preverili tipični testni vzorci, če pri njih ne najdemo resne hibe, se izvedejo tudi ostali testi. Ne izvaja se nobenih regresijskih testov.

B) Preverjanje programa v2.0

Program v2.0 bomo preverili s pregledom izvirne kode (stil kodiranja, skladnost s standardom) in testiranjem. Pripravljeni bodo določeni testni vzorci in postopki, ki jih bo natančneje definiral dokument Testni primeri. Preverjanje izvaja preverjevalec. Po preverjanju se izpolnijo pisna poročila o najdenih neustreznostih. Izvedejo se vsi testi (regresijsko testiranje).

Uporabljene bodo naslednje strategije (podroben opis je v prilogi tega dokumenta):

- prisotnost zahtev (Z)
- prepovedane vrednosti – za preverjanje robustnosti (R)
- mejne vrednosti (M)
- ugibanje napak oziroma nepravilnosti (U)

D2.4 NALOGE IN REZULTIRAJOČI DOKUMENTI (IZBRAN RAZVOJNI MODEL)

Pregled po produktih in aktivnostih

	Produkt	Planirana kompleksnost ¹	Dejanska kompleksnost ¹	Odgovorna oseba za produkt	V&V metoda	Odgovorna oseba za V&V	Način sporočanja o V&V	Opomba
D1	Naročnikove zahteve	1 stran	2 strani	P. Volker, D. Vinšek	splošni pregled	P. Volker, D. Vinšek	ustno	
D2	Plan projekta	6 strani	7 strani	P. Volker, D. Vinšek	splošni pregled	P. Volker, D. Vinšek	ustno/pisno	
D3	Sistemske specifikacije	10 strani	9 strani	P. Volker, D. Vinšek	splošni pregled	P. Volker, D. Vinšek	ustno/pisno	
	Program v1.0	600 LOC	700 LOC	P. Volker	testiranje + splošni pregled	P. Volker	pisno	
D4	Testni primeri	10 testnih primerov	10 testnih primerov	D. Vinšek	splošni pregled	D. Vinšek	ustno/pisno	
D5	Testno poročilo	5 strani	6 strani	D. Vinšek	splošni pregled	D. Vinšek	pisno	
D6	Načrtovalska dokumentacija	2 strani	5 strani	P. Volker	splošni pregled	P. Volker	ustno/pisno	
D7	Uporabniški priročnik	2 strani	6 strani	D. Vinšek	splošni pregled	D. Vinšek	pisno	
	Program v2.0	700 LOC	725 LOC	P. Volker	testiranje + splošni pregled	P. Volker	pisno	
	Kompleten produkt			P. Volker, D. Vinšek	testiranje + splošni pregled	P. Volker, D. Vinšek, naročnik	ustno/pisno	

Roki in stroški

	AKTIVNOST	Planiran rok	Dejanski rok	Planirani napor ¹	Planirani stroški	Dejanski napor	Dejanski stroški	Izvajalec	Odgovorna oseba
A1	Planiranje projekta in analiza zahtev	29. 10. 2020	28. 10. 2020	5	125	4	100	Deni Vinšek + Primož Volker	Primož Volker
A2	Načrtovanje	19. 11. 2020	19.11.2020	6	150	7	175	Deni Vinšek + Primož Volker	Primož Volker
A3	Implementacija programa v1.0	3. 1. 2021	3.1.2021	20	500	25	625	Primož Volker	Primož Volker
A4	Implementacija programa v2.0	14. 1. 2021	11.1.2021	10	250	1	25	Primož Volker	Primož Volker
A5	Načrtovanje testnih primerov	10. 12. 2020	17.12.2020	4	100	10	250	Deni Vinšek	Deni Vinšek
A6	Preverjanje programa v1.0	3. 1. 2021	12.1.2021	15	375	5	125	Deni Vinšek	Deni Vinšek
A7	Preverjanje programa v2.0	14. 1. 2021	17.1.2021	5	125	2	50	Deni Vinšek	Deni Vinšek
A8	Izdelava kompletne dokumentacije	21. 1. 2021	7.2.2021	14	350	9	225	Deni Vinšek + Primož Volker	Primož Volker
A9	Prezem	12.2.2021	12.2.2021	1	25	1	25	PROstor d.o.o.	Deni Vinšek + Primož Volker
	Skupaj napor - stroški			81	2000	63	1600	Deni Vinšek + Primož Volker	Primož Volker

Enota napora: človek-ure
Stroški enote napora: 25 EUR

¹ Navedi tudi ustrezno enoto (npr.: enota so ure ali stroški ali ljudje ali kombinacija).

D2.5 RESURSI

D2.5.1 Osebe (Kdo bo sodeloval, kakšna je njegova vloga, kakšne morajo biti njegove sposobnosti?)

	Oseba	AKTIVNOST	Vloga
P1	direktor podjetja PROstor d. o. o.	<ul style="list-style-type: none">• nadzor• prevzem	naročnik
P2	D. Vinšek	<ul style="list-style-type: none">• načrtovanje testnih primerov• testiranje	preverjevalec
P3	P. Volker	<ul style="list-style-type: none">• planiranje projekta• analiza zahtev• načrtovanje• implementacija programa v1.0• implementacija programa v2.0• izdelava načrtovalske dokumentacije• prevzem	razvojniki

D2.5.2 Potrebna programska orodja, knjižnice

orodje	namen, funkcija
Visual Studio 2017	kodiranje, odpravljanje neustreznosti
Microsoft Word	vodenje dokumentacije
Draw.io	izris grafov
Beležnica	ustvarjanje tekstovnih datotek

D2.5.3 Potrebna strojna oprema

orodje	namen, funkcija
dva računalnika	kodiranje, odpravljanje neustreznosti, vodenje dokumentacije
tiskalnik	izpis dokumentacije

D2.6 RAZDELITEV STROŠKOV

Glej točko D2.4.

D2.7 TERMINSKI PLAN PROJEKTA

	AKTIVNOST	ČASOVNA SKALA																															
		1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	
		1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16		
A1	Planiranje projekta in analiza zahtev	+	+	+	+	+																											
A2	Načrtovanje						+	+	+	+	+	+	+																				
A3	Implementacija programa v1.0													+	+	+	+	+	+	+	+	+											
A4	Implementacija programa v2.0																						+	+	+	+							
A5	Načrtovanje testnih primerov													+	+	+	+	*															
A6	Preverjanje programa v1.0																					+	+	+									
A7	Preverjanje programa v2.0																									+	+						
A8	Izdelava kompletne dokumentacije																											+	+				*
A9	Prevzem																													+	+		*
	DOKUMENT (skrajni rok)	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	
D1	Naročnikove zahteve			+																													
D2	Plan projekta						+																										
D3	Sistemske specifikacije										*	+																					
D4	Testni vzorci																						*	+									
D5	Testno poročilo																								*	+							
D6	Načrtovalska dokumentacija																									*	+			+		*	
D7	Uporabniški priročnik																													+	*		

Legenda:

Trajanje aktivnosti: + planiran čas, * dejansko porabljen čas

D2.8 POJMOVNIK

poj ^o m	razlaga
naročnik	PROstor d. o. o.

D2.9 PRILOGE

D2.9.1 Avtomatski način delovanja simulatorja

1. UKAZI ZA KRMILJENJE:

Temperatura

- Vkl^op grelca
- Izkl^op grelca

V nasprotnem primeru: Temperatura v prostoru je optimalna

Vlažnost

- Vkl^op vlažilca
- Izkl^op vlažilca

V nasprotnem primeru: Vlažnost prostora je optimalna

Osvetljenost

- Izkl^op luči in zatemnitev rolet (8000 lx+)
- Izkl^op luči (501 – 7999 lx)
- Prižig luči (101 – 499 lx)
- Prižig luči in odprtje rolet (10 – 100 lx)

V primeru, da je osvetljenost 500 lx: Osvetljenost prostora je optimalna

2. PRAVILNOSTNA TABELA:

T	V	O	IZHOD
0	0	0	Brez izvedenih ukazov
0	0	1	Ukaz za spremembo osvetljenosti
0	1	0	Ukaz za spremembo vlažnosti
0	1	1	Ukaz za spremembo vlažnosti + ukaz za spremembo osvetljenosti
1	0	0	$\Delta T < 10$; ukaz za spremembo temperature
			$\Delta T \geq 10$; ukaz za spremembo temperature + ukaz za spremembo vlažnosti, ki je posledica vpliva temperature na vlažnost prostora (+ 5%)
1	0	1	$\Delta T < 10$; ukaz za spremembo temperature + ukaz za spremembo osvetljenosti
			$\Delta T \geq 10$; ukaz za spremembo temperature + ukaz za spremembo vlažnosti, ki je posledica vpliva temperature na vlažnost prostora (+ 5%) + ukaz za spremembo osvetljenosti
1	1	0	$\Delta T < 10$; ukaz za spremembo temperature + ukaz za spremembo vlažnosti
			$\Delta T \geq 10$; ukaz za spremembo temperature + ukaz za spremembo vlažnosti, ki upošteva vlažnost kot posledico spremembe temperature (+ 5%)
1	1	1	$\Delta T < 10$; ukaz za spremembo temperature + ukaz za spremembo vlažnosti + ukaz za spremembo osvetljenosti
			$\Delta T \geq 10$; ukaz za spremembo temperature + ukaz za spremembo vlažnosti, ki upošteva vlažnost kot posledico spremembe temperature (+ 5%) + ukaz za spremembo osvetljenosti

D2.9.2 Napredni način delovanja simulatorja

1. UKAZI ZA KRMILJENJE:

Prezračevanje

- Vklp prezračevalnega sistema ($1500 \text{ ppm} \leq \text{vsebnost CO}_2 \leq 2500 \text{ ppm}$)
 - Odprtje okna ($1000 \text{ ppm} < \text{vsebnost CO}_2 < 1500 \text{ ppm}$)
 - Izklp prezračevalnega sistema ($\text{vsebnost CO}_2 < 400 \text{ ppm}$)
- V primeru, da je vsebnost CO₂ med 400 in 1000 ppm: Zrak v prostoru je optimalen

3. INFORMIRANJE:

Stanje razkužil

- Vsebnost razkužila na intervalu [700, 1000] ml: količina razkužila je optimalna
- Vsebnost razkužila na intervalu [690, 300] ml: količina razkužila je zadostna
- Vsebnost razkužila na intervalu [290, 100] ml: količina razkužila je nizka
- Vsebnost razkužila < 100 ml: količina razkužila je kritična

D2.9.3 Opisi uporabljenih strategij

1. Opis strategije: Prisotnost zahtev (Z)

1. Strategija je uporabna je v vseh primerih, kjer so znane specifikacije oziroma zahteve, med katerimi ni nobenih relacij. Predpostavka o napaki: določena zahteva ni implementirana. S to strategijo odkrivamo zahteve, ki niso implementirane. Razen zelo redkih izjem, ne bomo odkrili napačno implementiranih zahtev in zahtev, ki so po nepotrebnem implementirane.
2. Testirni model je seznam zahtev.
3. **Pravilo za načrtovanje testnih primerov:** Za vsako zahtevo tvori najmanj en testni primer. Vhodne podatke si poljubno izberi.
4. Z načrtovanjem testnih primerov lahko začnemo, ko so zahteve postavljene.
5. Testirna strategija je izčrpana, ko preverimo prisotnost vsake zahteve v seznamu.

2. Opis strategije za preverjanje robustnosti (R)

1. Strategija je uporabna je v vseh primerih, kjer je zahtevana robustnost in je možno tvoriti opis vhodne domene.
2. Predpostavka o nepravilnosti: program ni robusten, čeprav bi moral biti. S to strategijo ne bomo odkrili nepravilnosti, ki se pojavljajo pri procesiranju veljavnih podatkov.
3. Testirni model je opis vhodne domene.
4. **Pravilo za načrtovanje testnih primerov:** V vhodni domeni in identificiraj prepovedane razrede. Za vsak prepovedan razred tvori en testni primer.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je opisana vhodna domena.
6. Testirna strategija je izčrpana, ko smo pokrili vse neveljavne razrede v vhodni domeni. Zgornje število testnih primerov je enako številu neveljavnih razredov.

3. Opis strategije: ugibanje nepravilnosti (U)

1. Strategija je splošno uporabna.
2. Predpostavlja se, da je prisotna določena nepravilnost ali napaka.
3. Testirni model je seznam potencialnih nepravilnosti oziroma napak.
4. **Pravilo za načrtovanje testnih primerov:** Za vsako potencialno napako oziroma nepravilnost v seznamu tvorimo en testni primer, s katerim preverimo, ali je ta napaka/nepravilnost prisotna.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo pripravljen seznam.
6. Testirna strategija je izčrpana, ko smo pokrili celoten seznam. Zgornje število testnih primerov je enako številu napak oziroma nepravilnosti v seznamu.

4. Opis strategije: mejne vrednosti (M)

1. Strategija je splošno uporabna.
2. Predpostavka o nepravilnosti: vhodni podatki, ki se nahajajo v okolici ali pa točno na meji med veljavnim in neveljavnim območjem, se bodo nepravilno procesirali.
3. Testirni model je vhodna in izhodna domena.
4. **Pravilo za načrtovanje testnih primerov:** določi meje med veljavnimi in neveljavnimi podatki. Izberi vrednost točno na meji, malo nad in malo pod njo.
5. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo podatkovni slovar.
6. Testirna strategija je izčrpana, ko smo uporabili vse meje.

D3 SISTEMSKE SPECIFIKACIJE

D3.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument: Sistemske specifikacije	
verzija	1.0
datum	10. 11. 2020
lokacija	datoteka Projekt_Simulator_Vinsek_Volker.pdf
Odgovorni osebi	
	Primož Volker, Deni Vinšek
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve 2.0

D3.2 POVZETEK

Naročnik je 1. 10. 2020 predložil svoje zahteve, na podlagi katerih so bile izdelane te sistemske specifikacije.

Podjetje načrtuje centralni sistem za upravljanje prostorov, zato potrebujejo program v obliki simulatorja. Simulator mora v osnovi omogočati krmiljenje temperature, vlage in osvetljenosti prostora. Podpirati mora dva načina delovanja: testnega ter navadnega. Navaden se deli še na avtomatski in napredni (Covid-19) način. Podpirati mora tudi branje iz datoteke, kamor bodo vneseni vhodni podatki. Sistem mora samodejno prepoznati vneseno enoto in to pri izračunu upoštevati.

D3.3 ZAHTEVE GLEDE POSAMEZNIH KARAKTERISTIK

Kritične zahteve so označene s klicajem.

D3.3.1 Korektnost oziroma funkcionalnost

Glej točko D 3.7.

D3.3.2 Zanesljivost

Ker zahteve za izvedbo brezhibnega delovanja ni mogoče izvršiti, bomo program podrobno testirali po standardu CVVS 2-2000.

D3.3.3 Testabilnost

Program bo imel poleg navadnega tudi testni način, s katerim bo možno preveriti pravilno delovanje programa.

D3.3.4 Prenosljivost

Prenosljivost ni zahtevana, zahtevano je le delovanje na sistemu Windows.

D3.3.5 Prijaznost

Naročnik zahteva hiter in preprost program. Nobena zahteva ni podana na merljiv način. Izdelali bomo menijsko voden program z vgrajeno pomočjo za vse funkcije. V primeru napačnega oziroma nesmiselnega vnosa naj program deluje dalje in uporabnika opozori na napačen vnos.

D3.3.6 Razumljivost

Ni kvantitativnih in kvalitativnih zahtev.

D3.3.7 Varnost

Ni kvantitativnih in kvalitativnih zahtev.

D3.3.8 Vzdrževalnost

Program mora narejen tako, da ga lahko v prihodnosti vzdržuje in nadgrajuje tudi druga kvalificirana oseba. Mora biti dokumentiran v skladu s standardom.

D3.3.9 Zmogljivost

Ni kvantitativnih in kvalitativnih zahtev.

Program se bo preverjal na računalniku z i7 procesorjem 4. generacije in 16Gb pomnilnika. Program bo deloval podobno na vseh sodobnih računalnikih.

D3.4 OMEJITVE IN DRUGE ZAHTEVE

- 1) Za izvajanje programa mora zadoščati standardni PC.
- 2) Zagon programa:

```
Simulator.exe -t
```

Kretnica -t požene program v testnem režimu delovanja.

D3.5 OPIS SISTEMA

Opis funkcionalnosti je napravljen s pomočjo tipičnih vzorcev uporabe in diagramov.

D3.5.1 Tipični vzorci uporabe

TVZ_1. Testni režim

1. Zač. stanje: ukazna vrstica zagnana iz mape projekta, uporabnik pozna vhodne podatke
2. Iniciator: uporabnik želi preveriti delovanje programa
3. Opis dogodkov:
 - a. Vnos podatkov v tekstovno datoteko in shranjevanje
 - b. Poženemo program Simulator.exe
 - c. Program prebere podatke in jih obdela
 - d. Glede na vhodne podatke izvede ustrezen ukaz iz pravilnostne tabele
4. Končno stanje: izpisano sporočilo ob zaključku delovanja

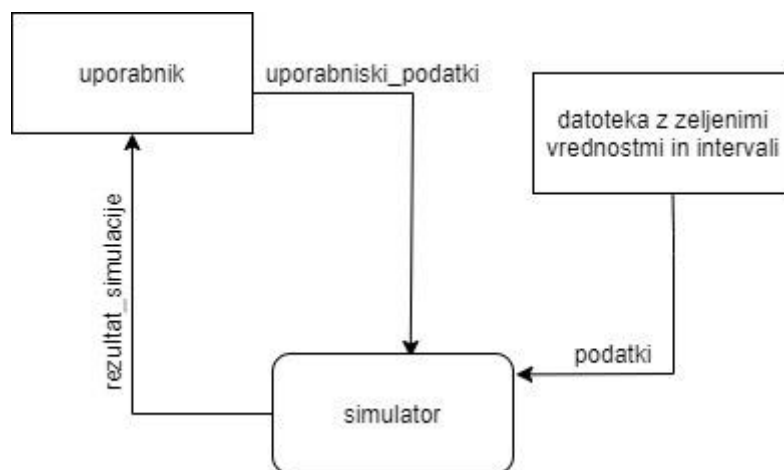
TVZ_2. Navadni režim

1. Zač. stanje: ukazna vrstica zagnana iz mape projekta, uporabnik pozna vhodne podatke
2. Iniciator: uporabnik želi avtomatsko prilagajanje stanja v prostoru
3. Opis dogodkov:
 - a. Vnos intervalov v tekstovno datoteko in shranjevanje
 - b. Poženemo program Simulator.exe
 - c. V začetnem meniju izberemo možnost za zagon simulacije
 - d. Program prebere interval in v obsegu sam določi vrednosti podatkov
 - e. Glede na vhodne podatke izvede ustrezen ukaz iz pravilnostne tabele
4. Končno stanje: izpisano sporočilo ob zaključku delovanja

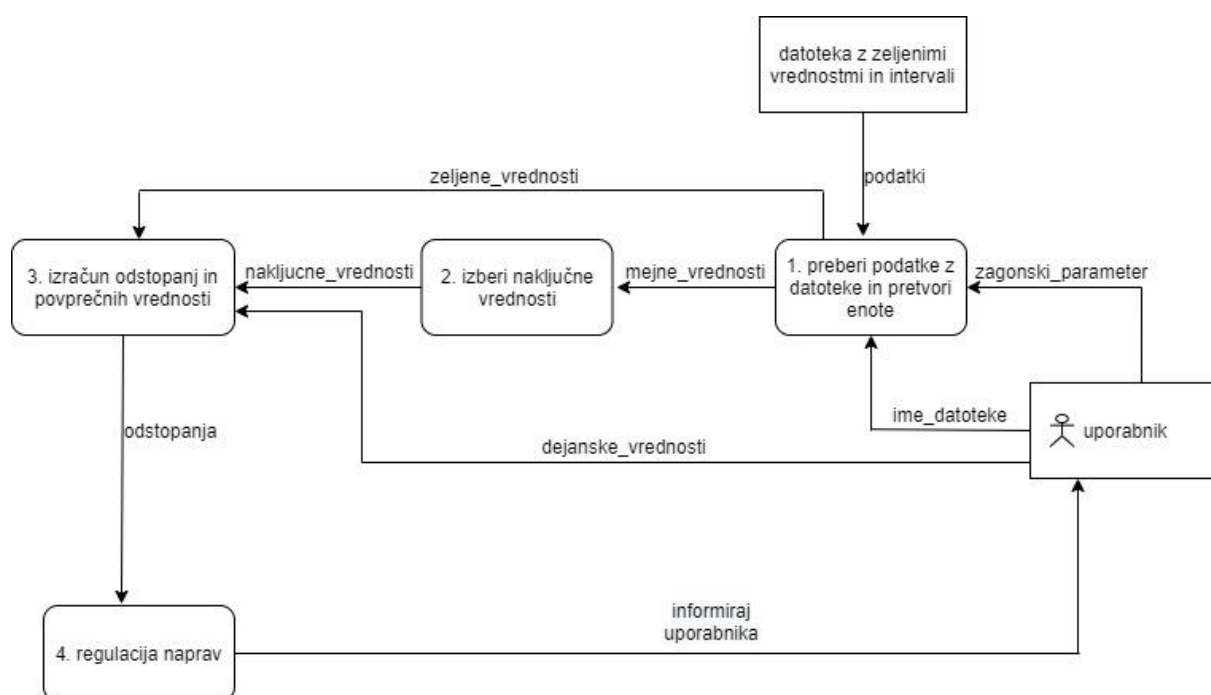
TVZ_3. Uporabnik zahteva pomoč

1. Zač. stanje: ukazna vrstica zagnana iz mape projekta
2. Iniciator: uporabnik potrebuje pomoč
3. Opis dogodkov:
 - a. Poženemo program Simulator.exe
 - b. V meniju izberemo pomoč (znak *).
4. Končno stanje: na ekranu je izpisana pomoč.

D3.5.2 Diagrami za opis sistema in podsistemov



Slika 1 – Nivo sistema (kontekstni nivo)



Slika 2 - Nivo podsistemov

OPIS PROCESOV

1. Preberi podatke z datoteke in pretvori enote

Glede na zagonski parameter proces uporabniku omogoči vnos dejanskih vrednosti v program ali pa prebere iz datoteke intervale, na katerih bodo vrednosti.

2. Izberi naključne vrednosti

Proces glede na podane intervale izbere naključne vrednosti temperature, vlage, osvetljenosti in vsebnosti razkužila.

3. Izračun odstopanj in povprečnih vrednosti

Proces izračuna odstopanja dejanskih vrednosti oz. naključno izbranih od željenih. Nato izračuna še povprečje izmerjenih vrednosti in povprečna odstopanja od željenih.

4. Regulacija naprav

Glede na izračunana odstopanja proces izvede ustrezne ukaze za regulacijo naprav. Uporabniku sporoči izvedene akcije in izpiše izmerjene ter izračunane vrednosti.

D3.6 OPIS PODATKOVNIH TOKOV IN TERMINATORJEV

D3.6.1 Podatkovni slovar za sliki 2 in 3

ime podatka (komponente so podane vsaka posebej)	atribut	tip	veljavno območje (domena)	Opomba
uporabniški_podatki	zagonski_parameter+ ime_datoteke+ dejanske_vrednosti	Znakovni niz, Cela števila	{-t,-c, },znakovni niz, {0,1,2,3,4,5,6,7,8,9}	Interval temperature: [10, 40] stopnja vlaznosti: [30,60] interval osvetljenosti: [10,10000]
rezultat_simulacije	informiraj_uporabnika +meni	Znakovni niz		
zagonski_parameter		Znakovni niz	{-t,-c, }	
ime_datoteke		Znakovni niz		
zeljenje_vrednosti		Cela števila	{0,1,2,3,4,5,6,7,8,9}	Interval temperature: [10, 40] stopnja vlaznosti: [30,60] interval osvetljenosti: [10,10000]
dejanske_vrednosti		Cela števila	{0,1,2,3,4,5,6,7,8,9}	Interval temperature: [10, 40] stopnja vlaznosti: [30,60] interval osvetljenosti: [10,10000]
podatki	mejne_vrednosti+ zeljene_vrednosti	Polje celih števil	{0,1,2,3,4,5,6,7,8,9}	Interval temperature: [10, 40] stopnja vlaznosti: [30,60] interval osvetljenosti: [10,10000]
odstopanja		Realna števila	Natančnost na 1 decimalko	MaxReal, MinReal
informiraj_uporabnika		Znakovni niz		
mejne_vrednosti		Polje celih števil	{0,1,2,3,4,5,6,7,8,9}	Interval temperature: [10, 40] stopnja vlaznosti: [30,60] interval osvetljenosti: [10,10000]
naključne_vrednosti		Cela števila	{0,1,2,3,4,5,6,7,8,9}	Interval temperature: [10, 40] stopnja vlaznosti: [30,60] interval osvetljenosti: [10,10000]

Konstante, ki so odvisne od prevajalnika: MaxReal = 3.4E38, MinReal = -3.4E38, MaxInteger = 2,147,483,647, MinInteger = -2,147,483,648. Navedene vrednosti veljajo na 32-bitnih prevajalnikih.

D3.6.2 Opis terminatorjev za sliki 2 in 3

ime terminatorja	opis
uporabnik	Uporabnik, ki dela s programom Simulator
datoteka z željenimi vrednostmi in intervali	Tekstovna datoteka. V njej so zapisani intervali in željene vrednosti v obliki: TEMPERATURA: vrednost [program zazna enoto samodejno, glede na vneseno vrednost. Možne enote so °C, K in °F] VLAZNOST: vrednost v obliki relativne vlažnosti [%] OSVETLJENOST: vrednost [lx] INTERVAL TEMPERATURE: [10, 40] - enota so °C STOPNJA VLAZNOSTI: [30,60] – relativna vlažnost v % INTERVAL OSVETLJENOSTI: [10,10000] – enota so lx

D3.7 PODROBEN OPIS IN INDEKSIRANJE FUNKCIJ IN DRUGIH ZAHTEV, KI JIH JE POTREBNO IMPLEMENTIRATI

Kritične so označene s klicajem.

- F1. Izpis verzije ob zagonu programa
Program mora ob zagonu vedno izpisati svojo verzijo.
- F2. Izpis pomoči
Obstajati mora možnost izpisa pomoči.
- F3. !! Branje podatkov iz datoteke
Program mora biti sposoben pridobiti podatke iz datoteke. V datoteki bodo intervali, znotraj katerih si bo izbral naključno vrednost.
- F4. Kontrola vhodnih podatkov (robustnost)
Program mora kontrolirati vhodne podatke in zavrniti neveljavne vnose v meniju. Opozoriti mora tudi na nepravilen zapis v datoteki.
- F5. Pretvorba podatkov
Program mora znati samodejno prepoznati ter pretvoriti temperaturne enote.
- F6. Izračun povprečnih odstopanj
Program mora izračunati in izpisati povprečna odstopanja izmerjenih vrednosti od željenih ter povprečja izmerjenih vrednosti.
- F7. ! Nastavitev posameznih naprav (izpis ukaza)
Program mora po pretvorbi podatkov ustrezno reagirati. Ob določenih vrednosti mora izvršiti ustrezen ukaz (npr. prižig luči ob osvetljenosti od 101 do 499 lx). Izpisati se mora, kaj je program prilagodil oz. obvestilo, če dodatni ukazi niso bili potrebni.
- Z1. Testni način delovanja.
Program mora podpirati testni režim delovanja, v katerem se izpisujejo dodatne informacije, vnos poteka preko tipkovnice.

D3.8 ZUNANJI VIDEZ

Program je bil pognan v normalnem režimu. Zaslona 4 in 5 spadata v testni režim. Pri testnem načinu se vrednosti vpišejo direktno v program, kjer se testira delovanje simulatorja.

izgled zaslona	opombe
zaslon 1 Simulator, verzija 1.0 (03.01.2021) ----- GLAVNI MENI ===== 1) Branje iz datoteke 2) Zagon simulacije *) Pomoc 0) Izhod iz programa >>	glavni meni V ukazni vrstici se moramo nahajati v mapi projekta. Simulator.exe Ta ukaz zažene program v navadnem načinu.
zaslon 2 Vnesi ime datoteke (ali * za izpis pomoci): podatki.txt Podatki so bili uspešno naloženi iz datoteke! Za vrnitev na glavni meni pritisni Enter.	Pri izbiri prve opcije lahko v program vnesemo ime datoteke s podatki.
zaslon 3 ===== SIMULACIJA ===== Vnesi stevilo simulacij (ali * za izpis pomoci): 2 Vnesi časovni razmik med simulacijami v sekundah: 3	Izbrana je bila druga opcija v glavnem meniju (datoteka je že naložena). Vnesemo koliko simulacij naj program izvede in časovni razmik med njimi.

<p>Zaslon 4</p> <p>TESTNI PODMENI =====</p> <p>1) Vnos dejanske temperature 2) Vnos dejanske vlaznosti 3) Vnos dejanske osvetljenosti 4) Vnos dejanske vsebnosti CO2 v zraku 5) Vnos dejanske kolicine razkuzila 0) Izhod na glavni meni</p> <p>>></p>	<p>Zaslon 4</p> <p>V testnem režimu je bila v glavnem meniju (zaslon 1) izbrana opcija Zagon simulacije.</p>
<p>zaslon 5</p> <p>=====</p> <p>SIMULACIJA =====</p> <p>Izmerjeni parametri -Parameter: x enota</p> <p>Željeni parametri -Parameter: x enota</p> <p>Izvedene operacije: -Izvedena operacija</p> <p>Za vrnitev na glavni meni pritisni Enter.</p>	<p>Zaslon 5</p> <p>V testnem režimu v podmeniju izberemo x opcijo.</p> <p>Izvede se simulacija glede na vneseno vrednost.</p>
<p>zaslon 6</p> <p>-----POMOC-----</p> <p>Pomoc pri zagonu simulacije =====</p> <p>Za pravilen zagon simulacije je potrebno v obeh primerih vnesti pozitivno stevilo do 1000. Zagon ni mozen z vnosom stevila 0 ali drugim vpisanim znakom, ki ni pozitivno celo stevilo.</p> <p>Za vrnitev na glavni meni pritisni Enter.</p>	<p>V zaslonu 3 je bil vnesen znak *. Izpiše se pomoč.</p>
<p>zaslon 7</p> <p>=====</p> <p>SIMULACIJA =====</p> <p>Simulacija 1:</p> <p>Izmerjeni parametri -Temperatura: 15C -Vlaznost: 55% -Osvetljenost: 110lx</p> <p>Željeni parametri -Temperatura: 20C -Vlaznost: 45% -Osvetljenost: 500lx</p> <p>Izvedene operacije: -Vklop grelca -Vklop vlazilca -Prizig luci</p> <p>-----</p>	<p>V zaslonu 3 je bilo vneseno število 2 za število simulacij, razmik med simulacijami pa 3 sekunde.</p> <p>Izvedeta se dve simulaciji. Izpišejo se izmerjeni in željeni parametri ter tudi izvedene operacije. Med simulacijami je razmik tri sekunde.</p> <p>Na koncu se izpišejo povprečne izmerjene vrednosti, ter povprečna odstopanja od željenih vrednosti.</p>

<p>Simulacija 2:</p> <p>Izmerjeni parametri</p> <ul style="list-style-type: none"> -Temperatura: 20C -Vlaznost: 40% -Osvetljenost: 2000lx <p>Zeljeni parametri</p> <ul style="list-style-type: none"> -Temperatura: 20C -Vlaznost: 45% -Osvetljenost: 500lx <p>Izvedene operacije:</p> <ul style="list-style-type: none"> -Temperatura v prostoru je optimalna -Vklop vlazilca -Izklop luci <p>-----</p> <p>Povprecne izmerjene vrednosti:</p> <ul style="list-style-type: none"> -Temperatura: 17.5C -Vlaznost: 47.5% -Osvetljenost: 305lx <p>Povprecna odstopanja od zeljenih vrednosti:</p> <ul style="list-style-type: none"> -Temperatura: 2.5C -Vlaznost: 7.5% -Osvetljenost: 945lx <p>Za vrnitev na glavni meni pritisni Enter.</p>	
---	--

Zaslon 8	
<p>-----POMOC-----</p> <p>Pomoc za glavni meni</p> <p>=====</p> <p>V tem meniju izbiras med stirimi moznostmi:</p> <p>1) Izbira z 1 prebere potrebne podatke iz tekstovne datoteke.</p> <p>2) Ce vpises stevilo 2 (in pritisnes enter) se ti odpre zaslon za dolocitev stevila simulacij, ter casovni razmik med njimi.</p> <p>*) Pri izbiri z znakom * se ti odpre ta zaslon za pomoc.</p> <p>0) Z niclo se izvajanje programa zakljuci.</p> <p>Za vrnitev na glavni meni pritisni Enter.</p>	V glavnem meniju je bila izbrana pomoc (*).
Zaslon 9	
<p>-----POMOC-----</p> <p>Pomoc pri branju podatkov iz datoteke</p> <p>=====</p> <p>Datoteka s podatki mora biti v pravilnem formatu, ime ne sme vsebovati sumnikov, na koncu pa je potrebno dodati koncnico .txt!</p> <p>Za vrnitev na glavni meni pritisni Enter.</p>	V zaslonu 2 je bila izbrana pomoc (*).

D3.9 OPIS FUNKCIJ, KI BODO NAJPREJ IMPLEMENTIRANE

- 1) Glavni meni
- 2) Branje podatkov iz datoteke
- 3) Pretvorba med temperaturnimi enotami
- 4) Izračun odstopanj in ostalih vrednosti
- 5) Izpis simulacij

D3.10 PREVZEMNI KRITERIJI

- 1) Program mora biti dokumentiran skladno s standardom CVVS-2/2000.
- 2) Program mora biti preverjen na najmanj 10 testnih primerih. Naročnik bo pripravil tri svoje testne primere, ki ne smejo pokazati na prisotnost večjih hib.

D3.11 POJMOVNIK

D4 TESTNI PRIMERI

D4.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument: Testni primeri	
verzija	1.0
datum	2.1.2021
lokacija	datoteka Projekt_Simulator_Vinsek_Volker.pdf
Odgovorna oseba	
	Deni Vinšek
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Sistemske specifikacije 1.0
	Plan projekta 1.0
	Naročnikove zahteve 2.0

D4.2 POVZETEK

Na podlagi naročnikovih zahtev in Plana projekta ter ostalih dokumentov, ki so navedeni v predhodni točki, je bil izdelan ta dokument, ki natančno definira postopek testiranja in testne primere. Dokument je uporaben za evalvacijo prototipa in za testiranje kompletnega programa. Rezultati testiranja so opisani v Poročilu o preverjanju.

D4.3 IDENTIFIKACIJA OBJEKTOV, NA KATERE SE TESTNI VZORCI NANAŠAJO

Testni primeri se nanašajo na program simulator.exe verzija 1 in 2.

D4.4 OPIS TESTNIH PRIMEROV

Glede na izbrane strategije (glej Plan projekta, točka Zagotavljanje kakovosti) smo tvorili 10 testnih primerov, ki so podrobneje opisani v prilogi. Pregled testnih primerov se nahaja v točki D4.6 (glej testno matriko), opis testnih modelov in primerov je v prilogi. V prilogi je tudi analiza vhodne in izhodne domene, ki jo potrebujemo pri strategiji mejnih vrednosti.

D4.5 OPIS TESTIRNEGA POSTOPKA

D4.5.1 Testiranje programa

V okolju MS Windows 10 testiramo tako, da najprej odpremo okno z ukaznim odzivnikom (»command prompt«), nakar s pomočjo ukaza CD skočimo v imenik, v katerem je program simulator.exe, ki ga bomo testirali. Testne primere vpisujemo interaktivno preko tipkovnice. Skoraj vsi testni primeri zahtevajo zunanjo datoteko podatki.txt v podimeniku s projektom.

D4.6 IDENTIFIKACIJA FUNKCIJ IN ZAHTEV TER TESTNA MATRIKA

Koda	Kratek opis	Kom- pleks- nost ¹	Stop- nja kriti- čnos- ti ²	Indikator pomem- bnosti ³	Število testov	T P 1 -1	T P 1 -2	T P 1 -3	T P 1 -4	T P 1 -5	T P 1 -6	T P 2 -1	T P 2 -2	T P 2 -3	T P 3 -1
F1	Izpis verzije ob zagonu programa	1	1	1(3%)	10(22%)	*	*	*	*	*	*	*	*	*	*
F2	Izpis pomoči	1	1	2(6%)	1(2%)		*								
F3	Branje podatkov iz datoteke	3	3	6(17%)	8(17%)	*		*	*	*	*		*	*	*
F4	Kontrola vhodnih podatkov (robustnost)	2	2	4(11%)	8(17%)	*		*	*	*		*	*	*	*
F5	Pretvorba podatkov	2	2	2(6%)	6(13%)	*		*	*	*			*		*
F6	Izračun povprečnih odstopanj	1	1	3(9%)	5(11%)	*			*	*	*				*
F7	Nastavitev posameznih naprav (izpis ukazov)	2	3	10(28%)	5(11%)	*			*	*	*				*
Z1	Testni način delovanja	2	3	7(20%)	3(7%)	*							*		*
	Skupaj			35(100%)	45 (100%)										
	pozitivni, negativni					P	P	P	P	P	P	N	N	N	N
	strategija⁴					Z	Z	Z	Z	Z	Z	R	R	R	M
	prioriteta pri izvrševanju⁵					3	2	3	2	K	2	2	2	1	3

¹ Kompleksnost : izberemo eno izmed metrik npr. LOC, če ni znana uporabimo subjektivno merilo (1 nizka kompleksnost, 2 srednja, 3 visoka).

² Stopnja kritičnosti: 3 zelo kritična, 2 kritična, 1 nepomembna

³ Indikator pomembnosti=število testov= stopnja kritičnosti x kompleksnost

⁴ Strategije: prisotnost zahtev (Z), mejna vrednost (M), enakovredni razredi (E), preverjanje robustnosti (R), ugotavljanje nepravilnosti (U), strukturno testiranje (S)

⁵ K = na koncu, x: ni pomembno vendar pred strukturnimi testi

D4.7 PRILOGA

D4.7.1 Opis¹ vhodne domene²

	ID	Ime podatka, stanja, dogodka	Tip podatka/Opis domene	Opomba
Uporabniški podatki	PAR	Zagonski parameter	Niz dveh znakov VR1=[-t,-c, brez] NR1=[ostali alfanumerični znaki v parametru]	
Uporabniški podatki	DAT	Ime datoteke	Znakovni niz z končnico .txt VR1=[npr. podatki.txt] NR1=[vse druge pripone .*]	Glej sintakso imen datotek za Windows oper. sistem
Dejanske / naključne vrednosti	TEMP	Temperatura	Celo število VR1=[10 <= x <= 40] NR1=x ne sme biti znak ali črka NR2=[40<x<10]	
Dejanske / naključne vrednosti	VL	Vlaga	Celo število VR1=[30 <= x <= 60] NR1=x ne sme biti znak ali črka NR2=[60<x<30]	
Dejanske / naključne vrednosti	OS	Osvetljenost	Celo število VR1=[10 <= x <= 10000] NR1=x ne sme biti znak ali črka NR2=[10000<x<10]	
Dejanske / naključne vrednosti	RAZ	Vsebnost razkužila	Celo število VR1=[0<=x<=100] NR1=x ne sme biti znak ali črka NR2=[100<x<0]	

Konstante, ki so odvisne od prevajalnika: MaxReal = 3.4E38, MinReal = -3.4E38, MaxInteger = 2,147,483,647, MinInteger = -2,147,483,648. Navedene vrednosti veljajo na 32-bitnih prevajalnikih.

¹ Začni s podatki, ki se nahajajo v podatkovnem slovarju.

² Opiši najmanj 10 najpomembnejših elementov (podatek, stanje, dogodek). VR: veljavna domena, NR: neveljavna domena

D4.7.2 Opis izhodne domene

	ID	Ime podatka, stanja, dogodka	Tip podatka/Opis domene	Opomba
Rezultat simulacije	IU	Izpis izvedenih ukazov	Znakovni niz/i	
Rezultat simulacije	IO	Izpis povprečnih odstopanj od željenih vrednosti	Znakovni niz/i + realno število/a natančno/a na 1 decimalko	
Pomoč	IP	Izpis pomoči	Znakovni niz/i	

D4.7.3 Priloga s konkretnimi testnimi primeri

Testni primeri so urejeni po strategijah.

Strategija Preverjanje prisotnosti zahtev

- TP1-1 Ali je prisotna najbolj kritična funkcionalnost? Ali je implementiran testni režim?
- TP1-2 Ali je vedno prisotna vgrajena pomoč?
- TP1-3 Ali program prebere podatke iz datoteke?
- TP1-4 Ali ima program napredni (Covid-19) način?
- TP1-5 Ali program izpiše povprečna odstopanja od željenih vrednosti?
- TP1-6 Je možno izbrati število iteracij in čas med simulacijami?

Strategija Preverjanje robustnosti

- TP2-1 Ali program pravilno odreagira na napačno izbiro v meniju?
 - TP2-1/1 Naključen znak
 - TP2-1/2 Neveljavna izbira (številka zunaj izbire v meniju)
- TP2-2 Ali program pravilno odreagira napačen vnos podatkov?
 - TP2-2/1 Decimalna vrednost
 - TP2-2/2 Negativna vrednost
 - TP2-2/3 Naključen znak poleg vrednosti
- TP2-3 Ali je program odporen na neveljavno ime datoteke?
 - TP2-3/1 Ime datoteke, ki ne obstaja
 - TP2-3/2 Napačna pripona datoteke

Strategija Mejne vrednosti

- TP3-1 Ali program pravilno odreagira na vnos napačnih dejanskih vrednosti?
 - TP-3-1/1 Vrednosti zunaj mejnih vrednosti
 - TP-3-1/2 Vrednosti enake mejnim vrednostim

Strategija 1: Preverjanje prisotnosti zahtev

4.7.3.1.1 TP1-1-Ali je prisotna najbolj kritična funkcionalnost? Ali je implementiran testni režim?

Testni primer:	TP1-1	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta, v imeniku je datoteka podatki.txt	Strategija:	Preverjanje prisotnosti zahtev Tipični scenarij uporabe, pozitivni testni primer
Namen	Ali je prisotna najbolj kritična funkcionalnost? Ali je implementiran testni režim?	Opomba:	Mejne vrednosti so podane v datoteki. Testni primer je primeren za demonstracijo delovanja programa.
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Odvisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe -t		Izpisana pravilna verzija programa. Potem se javi glavni meni (zaslon 1). Ker je v testnem režimu, mora biti zaslon oštevilčen.	Stanje po zagonu programa	
2	Izberem Branje iz datoteke	1	Vnos imena datoteke		
3	Vnesem ime datoteke	podatki.txt	Potrditev o branju z datoteke		
4	Nadaljujem	Enter	Podmeni		
5	Izberem Zagon simulacije	2	Podmeni za vnos dejanskih vrednosti		
6	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
7	Vnesem dejansko temperaturo	10	Zaslon 5 z izmerjenim parametrom 15C, željenim parametrom 21C in izvedeno operacijo vklopa grelca + sprememba vlažnosti +5%.	Program mora pravilno regulirati napravo	
8	Nadaljujem	Enter	Podmeni Zagon simulacije		
9	Izberem vnos dejanske vlažnosti	2	Poziv za vnos vrednosti		
10	Vnesem dejansko vlažnost	40	Zaslon 5 z izmerjenim parametrom 40%, željenim parametrom 45% in vklopom vlažilca	Program mora pravilno regulirati napravo	
11	Nadaljujem	Enter	Podmeni Zagon simulacije		
12	Izberem vnos dejanske osvetljenosti	3	Poziv za vnos vrednosti		
12	Vnesem dejansko osvetljenost	300	Zaslon 5 z izmerjenim parametrom 300lx, željenim parametrom 8000lx in izvedeno operacijo vklopa luči.	Program mora pravilno regulirati napravo	

13	Nadaljujem	Enter	Podmeni Zagon simulacije		
14	Izberem vnos dejanske vsebnosti CO2 v zraku	4	Poziv za vnos vrednosti		
15	Vnesem dejansko vsebnost CO2 v zraku	230	Zaslon 5 z izmerjenim parametrom 230ppm, željenim parametrom me 400 in 1000ppm in izvedeno operacijo izklopa prezračevalnega sistema.	Program mora pravilno regulirati napravo - Vkllop prezračevalnega sistema (1500ppm<=CO2<=2500 ppm) - Odprtje okna (1000 ppm<CO2<1500 ppm) - Izklop prezračevalnega sistema (CO2 < 400 ppm) V primeru, da je vsebnost CO2 med 400 in 1000 ppm: Zrak v prostoru je optimalen	
16	Nadaljujem	Enter	Podmeni Zagon simulacije		
17	Izberem vnos dejanske količine razkužila	5	Poziv za vnos vrednosti		
18	Vnesem dejansko količino razkužila	500	Zaslon 5 z izpisano informacijo o zadostni količini razkužila.	- Vsebnost razkužila na intervalu [700, 1000] ml: količina razkužila je optimalna - Vsebnost razkužila na intervalu [300, 690] ml: količina razkužila je zadostna - Vsebnost razkužila na intervalu [290, 100] ml: količina razkužila je nizka - Vsebnost razkužila < 100 ml: količina razkužila je kritična	
19	Nadaljujem	Enter	Podmeni Zagon simulacije		
20	Izhod iz podmenija	0	Glavni meni (zaslon 1)		
21	Končam z uporabo	0	Command prompt		

4.7.3.1.2 TP1-2 Ali je vedno prisotna vgrajena pomoč?

Testni primer:	TP1-2	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta, v imeniku je datoteka podatki.txt	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer. V vsakem menuju zahtevaj izpis pomoči.
Namen	Ali je vedno prisotna vgrajena pomoč?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Odkvisnost od drugih testov:	Ni nobene odkvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe		Glavni meni (zaslon 1)	Stanje po zagonu programa	
2	Zahtevam izpis pomoči	*	Zaslon 8		
3	Nadaljujem	Enter	Glavni meni		
4	Izberem Beri iz datoteke	1	Vnos imena datoteke		
5	Zahtevamo izpis pomoči	*	Zaslon 9		
6	Nadaljujem	Enter	Glavni meni		
7	Izberem Beri iz datoteke	1	Vnos imena datoteke		
8	Vnesem ime datoteke	Podatki.txt	Potrditev o branju		
9	Nadaljujem	Enter	Glavni meni		
10	Izberem Zagon simulacije	2	Program želi, da vnesemo št. simulacij in časovni razmik med njimi.		
11	Zahtevam izpis pomoči	*	Zaslon 6		
12	Nadaljujem	Enter	Glavni meni		
13	Zapustim program	0	Command prompt		

4.7.3.1.3 TP1-3 Ali program prebere podatke iz datoteke?

Testni primer:	TP1-3	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta, v imeniku je datoteka podatki.txt	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer
Namen	Ali prebere mejne vrednosti in željene vrednosti iz datoteke?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel core i7 4790k 4.6GHz	Odpisnost od drugih testov:	Ni nobene odpisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe		Glavni meni (zaslon 1)	Stanje po zagonu programa	
2	Izberem Beri iz datoteke	1	Vnos imena datoteke		
3	Vnesem ime datoteke	podatki.txt	Izpis potrditve o branju oz. napaka in nato glavni meni (zaslon 1)	Opazujemo napako pri branju it datoteke	
4	Nadaljujem	Enter	Glavni meni		
5	Končam z uporabo	0	Command prompt		

4.7.3.1.4 TP1-4 Ali ima program napredni (COVID-19) način?

Testni primer:	TP1-4	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta, v imeniku je datoteka podatki.txt	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer
Namen	Ali ima program napredni (COVID-19) način?	Opomba:	Poženemo ga z parametrom -c pri zagonu
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Odpisnost od drugih testov:	

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe -c		Glavni meni (zaslon 1)	Opazujemo, če je funkcija implementirana.	
2	Izberem Beri iz datoteke	1	Vnos imena datoteke		
3	Vnesem ime datoteke	podatki.txt	Izpis potrditve o branju oz. napaka in nato glavni meni (zaslon 1)		
4	Nadaljujem	Enter	Glavni meni		
5	Izberem Zagon simulacije	2	Program želi, da vnesemo št. simulacij in časovni razmik med njimi.		
6	Vpišem neko število simulacij	6		Na vsakih 5 simulacij izpis o stanju razkužila	
7	Nadaljujem	Enter	Vnos časovnega zamika		
8	Vpišem časovni razmik	2			
9	Nadaljujem	Enter	Izpis rezultat simulacije, ki poleg podatkov iz navadne simulacije (zaslon 7) vsebuje tudi informacijo o stanju razkužila in vsebnosti CO2.		
10	Končam z uporabo	0	Command prompt		

4.7.3.1.5 TP1-5 Ali program izpiše povprečna odstopanja od željenih vrednosti?

Testni primer:	TP1-5	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta, v imeniku je datoteka podatki.txt	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer
Namen	Ali program izpiše povprečna odstopanja od željenih vrednosti?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Odvisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe		Glavni meni (zaslon 1)		
2	Izberem Beri iz datoteke	1	Vnos imena datoteke		
3	Vnesem ime datoteke	podatki.txt	Izpis potrditve o branju oz. napaka in nato glavni meni (zaslon 1)		
4	Nadaljujem	Enter	Glavni meni		
5	Izberem Zagon simulacije	2	Program želi, da vnesemo št. simulacij in časovni razmik med njimi.		
6	Vpišem neko število simulacij	2			
7	Nadaljujem	Enter	Vnos časovnega zamika		
8	Vpišem časovni razmik	3			

9	Nadaljujem	Enter	<pre> ===== SIMULACIJA ===== Simulacija 1: Izmerjeni parametri -Temperatura:15C -Vlaznost:15% -Osvetljenost:110lx Zeljeni parametri -Temperatura:21C -Vlaznost:45% -Osvetljenost:500lx Izvedene operacije: -Vklop grelca -Vklop vlazilca -Izklop luci ----- Simulacija 2: Izmerjeni parametri -Temperatura:21C -Vlaznost:40% -Osvetljenost:2000lx Zeljeni parametri -Temperatura:21C -Vlaznost:45% -Osvetljenost:500lx Izvedene operacije: -Vklop vlazilca -Izklop luci ----- </pre>	Izpisana pravilna povprečna odstopanja ter povprečne izmerjene vrednosti	
---	------------	-------	--	--	--

9			Povprecne izmerjene vrednosti: -Temperatura:16,5C -Vlaznost:27,5% -Osvetljenost:1055lx Povprecna odstopanja od zeljenih vrednosti: -Temperatura:4,5C -Vlaznost:17,5% -Osvetljenost:555lx Za vrnitev na glavni meni pritisni Enter.		
10	Nadaljujem	Enter	Glavni meni		
11	Končam z uporabo	0	Command prompt		

4.7.3.1.6 TP1-6 Je možno izbrati število iteracij in čas med simulacijami?

Testni primer:	TP1-6	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta, v imeniku je datoteka podatki.txt	Strategija:	Preverjanje prisotnosti zahtev, pozitivni testni primer
Namen	Ali je možno izbrati število iteracij in čas med simulacijami?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Odpisnost od drugih testov:	Ni nobene odpisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe		Glavni meni (zaslon 1)		
2	Izberem Beri iz datoteke	1	Vnos imena datoteke		
3	Vnesem ime datoteke	podatki.txt	Izpis potrditve o branju oz. napaka in nato glavni meni (zaslon 1)		
4	Nadaljujem	Enter	Glavni meni		
5	Izberem Zagon simulacije	2	Program želi, da vnesemo št. simulacij in časovni razmik med njimi.		
6	Vpišem neko število simulacij	10			
7	Nadaljujem	Enter	Vnos časovnega zamika		
8	Vpišem časovni razmik	3			
9	Nadaljujem	Enter	Izpis rezultat simulacije (zaslon 4).	Izvedeno mora biti 10 simulacij z razmikom 3 sekund	
10	Nadaljujem	Enter	Glavni meni		
11	Končam z uporabo		Command prompt		

D4.7.3.2 Strategija 2: Preverjanje robustnosti

4.7.3.2.1 TP2-1 Ali program pravilno odreagira na napačno izbiro v meniju?

Testni primer:	TP2-1	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Windows Command prompt v imeniku projekta	Strategija:	Preverjanje robustnosti pri vnosu
Namen	Kako program odreagira na napačen pri izbiri v meniju?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Ovisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vtipkam Simulator.exe		Glavni meni (zaslona 1)		
2	Vnesem napačno izbiro	u	Izpis opozorila o napaki pri izbiri	Opazujemo morebitno nepričakovano prenehanje delovanja programa	
3	Vnesem napačno izbiro	5	Izpis opozorila o napaki pri izbiri	Opazujemo morebitno nepričakovano prenehanje delovanja programa	
4	Vnesem napačno izbiro	(nič)	Izpis opozorila o napaki pri izbiri		
5	Izhod	0	Command prompt		

4.7.3.2.2 TP2-2 Ali program pravilno odreagira na napačen vnos podatkov?

Testni primer:	TP2-2	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Glej TP1-1 korak 5	Strategija:	Preverjanje robustnosti pri vnosu
Namen	Kako program odreagira na vnos napačnih dejanskih vrednosti?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Odvisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
2	Vnesem napačno vrednost	23.8	Program javi napako pri vnosu	Opazujemo za nepričakovano ustavitev programa	
3	Nadaljujem	Enter			
4	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
5	Vnesem napačno vrednost	-23	Program javi napako pri vnosu	Opazujemo za nepričakovano ustavitev programa	
6	Nadaljujem	Enter			
7	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
8	Vnesem napačno vrednost	22u	Program javi napako pri vnosu	Opazujemo za nepričakovano ustavitev programa	
9	Nadaljujem	Enter			
10	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
11	Vnesem pravilno vrednost	15			
12	Nadaljujem	Enter	Zaslon 5 z izmerjenim parametrom 15, željenim parametrom 21 in vklopom grelca.		
13	Nadaljujem	Enter	Podmeni Zagon simulacije		
14	Izhod	0	Glavni meni (zaslon 1)		

4.7.3.2.3 TP2-3 Ali je program odporen na neveljavno ime datoteke?

Testni primer:	TP2-3	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Glej TP1-3 korak 2	Strategija:	Preverjanje robustnosti pri branju iz datoteke
Namen	Ali je program odporen na ime datoteke, ki ne obstaja?	Opomba:	
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel Core i7 4790k 4.6GHz	Ovisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Vnesem napačno ime datoteke	nekaj.txt	Izpis napaka pri vnosu imena datoteke	Opazujemo napake pri branju oz. morebitno prenehanje delovanja programa	
2	Nadaljujem	Enter			
3	Izberem Beri iz datoteke	1	Vnos imena datoteke		
4	Vnesem napačno ime datoteke	podatki.jpg	Izpis napaka pri vnosu imena datoteke in zahteva za ponoven vnos	Opazujemo napake pri branju oz. morebitno prenehanje delovanja programa	
5	Nadaljujem	Enter			
6	Vnesem pravilno ime datoteke	podatki.txt	Izpis o uspešnem branju iz datoteke		
7	Nadaljujem	Enter	Glavni meni		
8	Izhod	0	Command prompt		

D4.7.3.3 Strategija 3: Mejne vrednosti

4.7.3.3.1 TP3-1 Vnos dejanskih vrednosti zunaj mejnih vrednosti

Testni primer:	TP3-1	TESTNI PRIMER	Verzija:	V 1.0	Projekt: Simulator
Garnitura:			Avtor:	Deni Vinšek	

Začetno stanje	Glej TP1-1 korak 5	Strategija:	Preverjanje robustnosti pri vnosu
Namen	Kako program odreagira na vnos vrednosti zunaj mejnih vrednosti, vrednosti enake mejnim?	Opomba:	Mejne vrednosti temperature: [10,40]
Konfiguracija sistema:	Windows 10, 16GB RAM, Intel core i7 4790k 4.6GHz	Odvisnost od drugih testov:	Ni nobene odvisnosti

korak	akcija / ime vhodne spremenljivke	vrednost	pričakovana reakcija/vrednost	opomba/odpovedna krit. funk.	opomba za testno poročilo
1	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
2	Vnesem vrednost nad mejo	41	Program javi napako o vrednosti izven meja	Opazujemo napake pri branju oz. morebitno prenehanje delovanja programa	
3	Nadaljujem	Enter			
4	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
5	Vnesem vrednost na zgornji meji	40	Rezultat simulacije (zaslon 5)	Opazujemo napake pri branju oz. morebitno prenehanje delovanja programa	
6	Nadaljujem	Enter	Podmeni Zagon simulacije (zaslon 4)		
7	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
8	Vnesem vrednost pod mejo	9	Program javi napako o vrednosti izven meja	Opazujemo napake pri branju oz. morebitno prenehanje delovanja programa	
9	Izberem vnos dejanske temperature	1	Poziv za vnos vrednosti		
10	Vnesem vrednost na spodnji meji	10	Rezultat simulacije (zaslon 5)	Opazujemo napake pri branju oz. morebitno prenehanje delovanja programa	
11	Nadaljujem	Enter	Podmeni Zagon simulacije(zaslon 4)		
12	Izhod	0	Glavni meni (zaslon 1)		

D5 POROČILO O PREVERJANJU

D5.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument <u>Poročilo o preverjanju</u>	
verzija	1.0
datum	7.1.2021
lokacija	datoteka Projekt_Simulator_Vinsek_Volker.pdf
Odgovorna oseba	
	Deni Vinšek
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Sistemske specifikacije 1.0
	Plan projekta 1.0
	Naročnikove zahteve 2.0
	Testni primeri 1.0

D5.2 POVZETEK

Program smo preverili skladno z testnimi primeri 1.0 in načrtom zagotavljanja kakovosti opisanem v planu projekta (točka D2.3). Uporabili smo 3 strategije preverjanja (točka D4.7.3) in izvedli 10 testnih primerov. Vsak test je bil izveden v skladu s standardom CVVS 2-2000, najdene nepravilnosti pa smo opisali v poročilih o nepravilnostih (D5.4). Ugotovljene nepravilnosti so bile pisno sporočene razvijalcu programske opreme.

V programu so bile implementirane vse željene zahteve oz. ni bilo najdenih ne implementiranih funkcij pri testiranju. V prvi verziji so bile odkrite 4 nepravilnosti. Od tega smo vse štiri takoj odpravili. V dveh primerih gre za isto nepravilnost najdeno v dveh različnih delovanjih programa. Obe se pojavita če se simulacija izvede več kot enkrat. Pri testiranju verzije 2.0 nismo našli napak.

D5.2.1 Pregled izvirne kode (verzija 1.0)

Program obsega 725 vrstic kode in 172 komentarjev. Izvorna koda je bila pregledana in je v skladu s standardom cV&Vs. Testiranje je potekalo na računalniku PC s sistemom Windows 10 (različica 1909) s procesorjem Intel core i7 4790k (4,6Ghz) in 16Gb pomnilnika. Testirali smo dve verziji programa Simulator.exe (V1.0 in V2.0).

D5.3 POROČILO O NAJDENIH NEPRAVILNOSTIH

Pregledni seznam vseh opaženih neustreznosti:

Nepri- vilnost štev.	Testni primer	Opis nepravilnosti	Resnost	Nepравilnost je bila odkrita v verziji	Končni status nepravilnosti	Opomba
1/1	TP1-1	Izpisano nepotrebno opozorilo	1	V 1.0	Odpravljeno v verziji 2.0	
1/2	TP1-4	Napačen izračun povprečnih vrednosti (napredni način)	2	V 1.0	Odpravljeno v verziji 2.0	
1/3	TP1-5	Napačen izračun povprečnih vrednosti (navaden način)	2	V 1.0	Odpravljeno v verziji 2.0	
2/1	TP2-2	Napačno implementiran vnos decimalnih, napačnih vrednosti	2	V 1.0	Odpravljeno v verziji 2.0	

Odkrili smo skupaj 4 različne nepravilnosti na 725 vrstic izvirne kode.

D5.4 PRILOGE

D5.4.1 Poročila o neustreznostih



Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

POROČILO O NEPRAVILNOSTI

Forma: ONEU V3.1s

1) PROJEKT: SIMULATOR 2) NEPRAVILNOST ŠTEV.: 1/1

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: podatki.txt

6) VRSTA NEPRAVILNOSTI (1-3):
☒ 1 Nepravilno implementirano
2 Ni implementirano
3 Ni bilo zahtevano

7) RESNOST NEPRAVILNOSTI (1-3):
☒ 1 Nepomembna
2 - Resna
3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: zastan 4 - testni podmeni

9) KRATEK OPIS NEPRAVILNOSTI: Ob izhodu na glavni meni izpisano opozorilo o neveljavnem številu.

10) JE PONOVLJIVOST ZAGOTOVLJENA ((da/ne)?: ne)

11) PREDVIDENA POGOSTOST POJAVLJANJA: 1 zelo pogosto, 2 občasno, ☒ 3 redko

12) TESTNI PRIMER/POSTOPEK: GLEJ TP1-1

13) PRILOGE: /

14) POROČEVALEC: D. Vinšek 15) DATUM: 9.1.2021

16) ODGOVORNA OSEBA: P. Volker Izpolni vodja

17) PRIORITETA: 1 Takoj odpraviti ☒ 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1 19) Podpis vodje Volker
(glej polje 20)

20) NEPRAVILNOST JE (1-9): Izpolni razvojna skupina
☒ 1 Odpravljena 4 Zavrnjena 7 Ni več aktualna
2 Prestavljena (hiba) 5 Preklicana 8 Duplikat
3 Neponovljiva 6 Ignorirana 9 Potrebne so dodatne inf.

21) OPOMBA /

22) NEPRAVILNOST ODPRASIL P. Volker 23) DATUM 11.1.2021

24) POPRAVILO PREVERIL D. Vinšek 25) DATUM 12.1.2021



POROČILO O NEPRAVILNOSTI

Forma: ONEU V3.1s

1) PROJEKT: SIMULATOR 2) NEPRAVILNOST ŠTEV.: 1/2

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: podatki.txt

6) VRSTA NEPRAVILNOSTI (1-3):
☒ 1 Nepravilno implementirano
☐ 2 Ni implementirano
☐ 3 Ni bilo zahtevano

7) RESNOST NEPRAVILNOSTI (1-3):
☒ 1 - Nepomembna
☐ 2 - Resna
☐ 3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZPIS POVPREČNIH VREDNOSTI - NAPREDNI NAČIN

9) KRATEK OPIS NEPRAVILNOSTI: PROGRAM NAPAČNO IZRAČUNA
POVPREČNE IZMERJENE VREDNOSTI, POVPREČNA
ODSTOPANJA OD ŽELJENIH VREDNOSTI

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)? da

11) PREDVIDENA POGOSTOST POJAVLJANJA: 1 zelo pogosto, 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: GLEJ TP1-4

13) PRILOGE: /

14) POROČEVALEC: D. VINŠEK 15) DATUM: 9.1.2021

16) ODGOVORNA OSEBA: P. Volker Izpolni vodja

17) PRIORITETA: ☒ 1 Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1 19) Podpis vodje: Volker
(glej polje 20)

20) NEPRAVILNOST JE (1-9): Izpolni razvojna skupina
☒ 1 Odpravljena
☐ 2 Prestavljena (hiba)
☐ 3 Neponovljiva
☐ 4 Zavržena
☐ 5 Preklicana
☐ 6 Ignorirana
☐ 7 Ni več aktualna
☐ 8 Duplikat
☐ 9 Potrebne so dodatne inf.

21) OPOMBA: /

22) NEPRAVILNOST ODPRAVIL: P. Volker 23) DATUM: 11.1.2021

24) POPRAVILO PREVERIL: D. Vinšek 25) DATUM: 12.1.2021



POROČILO O NEPRAVILNOSTI

Forma: ONEU V3.1s

1) PROJEKT: SIMULATOR 2) NEPRAVILNOST ŠTEV.: 1/3

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: podatki.txt

6) VRSTA NEPRAVILNOSTI (1-3):
☒ Nepravilno implementirano
2 Ni implementirano
3 Ni bilo zahtevano

7) RESNOST NEPRAVILNOSTI (1-3):
1 - Nepomembna
☒ 2 - Resna
3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: IZPIS POVPREČNIH VREDNOSTI - NAVADEN NAČIN

9) KRATEK OPIS NEPRAVILNOSTI: PROGRAM NAPAČNO IZRAČUNA
POVPREČNE IZMERJENE VREDNOSTI, POVPREČNA
ODSTOPANJA OD ŽELJENIH VREDNOSTI

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)? ☒ da ☐ ne

11) PREDVIDENA POGOSTOST POJAVLJANJA: 1 zelo pogosto, ☒ 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: GLEJ TP1-5

13) PRILOGE: /

14) POROČEVALEC: P. VINŠEK 15) DATUM: 9.1.2021

16) ODGOVORNA OSEBA: P. Volker Izpolni vodja

17) PRIORITETA: ☒ 1 Takoj odpraviti 2 Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1 19) Podpis vodje: Volker
(glej polje 20)

20) NEPRAVILNOST JE (1-9):
Izpolni razvojna skupina

<input checked="" type="radio"/> 1 Odpravljena	4 Zavrnjena	7 Ni več aktualna
2 Prestavljena (hiba)	5 Preklicana	8 Duplikat
3 Neponovljiva	6 Ignorirana	9 Potrebne so dodatne inf.

21) OPOMBA: /

22) NEPRAVILNOST ODPRAVIL: P. Volker 23) DATUM: 11.1.2021

24) POPRAVILO PREVERIL: D. Vinšek 25) DATUM: 12.1.2021



POROČILO O NEPRAVILNOSTI

Forma: ONEU V3.1s

1) PROJEKT: SIMULATOR 2) NEPRAVILNOST ŠTEV.: 2/1

3) TESTNI OBJEKT: PROGRAM 4) VERZIJA: 1.0

5) KONFIGURACIJA: podatki.txt

6) VRSTA NEPRAVILNOSTI (1-3):
① Nepravilno implementirano
2 Ni implementirano
3 Ni bilo zahtevano

7) RESNOST NEPRAVILNOSTI (1-3):
1 - Nepomembna
② - Resna
3 - Zelo pomembna

8) LOKACIJA NEPRAVILNOSTI: Vnos dejanske temperature - testni način

9) KRATEK OPIS NEPRAVILNOSTI: OB VNOSU DECIMALNE VREDNOSTI
PROGRAM SPREJME VREDNOST IN JO UPOŠTEVA BREZ
DECIMALKE (NAMENSKO ?) + ZAMAKNjen IZPIS TESTNEGA
PODMENIJA.

OB VNOSU ČRKE POLEG VREDNOSTI SE PROGRAM VRNE
NA PODMENI BREZ KAKŠNEGA OPOZORILA

10) JE PONOVLJIVOST ZAGOTOVLJENA (da/ne)?

11) PREDVIDENA POGOSTOST POJAVLJANJA: ① zelo pogosto, 2 občasno, 3 redko

12) TESTNI PRIMER/POSTOPEK: GLEJ TP 2-2

13) PRILOGE: /

14) POROČEVALEC: D. VINŠEK 15) DATUM: 9.1.2021

16) ODGOVORNA OSEBA P. Volker Izpolni vodja

17) PRIORITETA: 1 Takoj odpraviti ② Čim bo mogoče

18) KONČNI STATUS NEPRAVILNOSTI 1 19) Podpis vodje Volker
(glej polje 20)

Izpolni razvojna skupina

20) NEPRAVILNOST JE (1-9):

① Odpravljena	4 Zavrnjena	7 Ni več aktualna
2 Prestavljena (hiba)	5 Preklicana	8 Duplikat
3 Neponovljiva	6 Ignorirana	9 Potrebne so dodatne inf.

21) OPOMBA /

22) NEPRAVILNOST ODPRAVIL P. Volker 23) DATUM 11.1.2021

24) POPRAVILO PREVERIL D. Vinšek 25) DATUM 12.1.2021

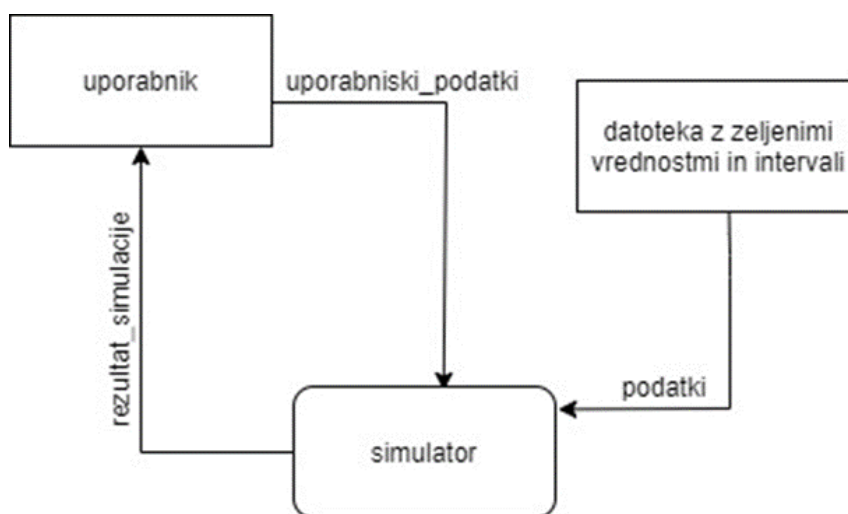
D6 NAČRTOVALSKA DOKUMENTACIJA

D6.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z DRUGIMI DOKUMENTI

Dokument	
<u>Načrtovalska dokumentacija</u>	
verzija	1.0
datum	4. 2. 2021
lokacija	datoteka Projekt_Simulator_Vinsek_Volker.pdf
Odgovorna oseba	
	Primož Volker
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve 2.0
	Plan projekta 1.0
	Sistemske specifikacije 1.0
	Uporabniški priročnik 1.0
	Testni primeri 1.0

D6.2 POVZETEK IZ SPECIFIKACIJ

D6.2.1 Kontekstni nivo



D6.2.2 Datoteke, ki jih uporablja uporabnik

Uporabnik za delovanje programa potrebuje tekstovno datoteko, v kateri so zapisani podatki v določenem vrstnem redu. Struktura datoteke mora ostati zmeraj enaka, uporabnik lahko spreminja le vrednosti. Teh datotek je lahko več. Primer imena datoteke: *podatki.txt*.

Za zagon programa uporabnik potrebuje izvršilno datoteko *Simulator.exe*. V tej datoteki se nahaja koda programa, spreminjanje te datoteke ni dovoljeno.

D6.2.3 Zagon programa

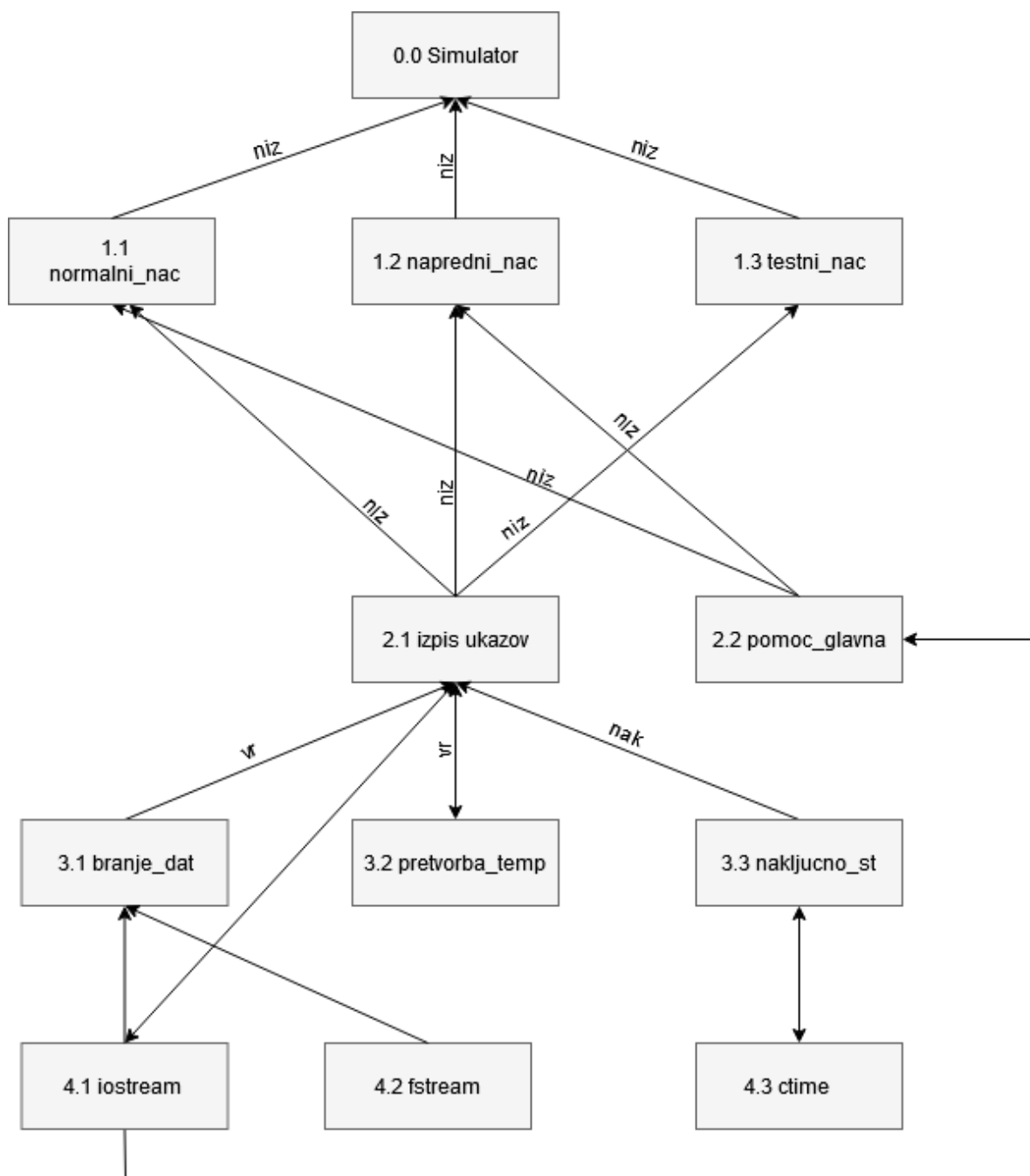
Zagon programa poteka preko ukazne vrstice. Nahajati se moramo v mapi projekta, nato vtipkamo *Simulator.exe* za normalni način. Doda se lahko tudi parameter *-t* za testni način ali *-c* za napredni način. Zagon normalnega načina je mogoč tudi z dvoklikom na ikono ali bližnjico.

D6.2.4 Datoteke, ki jih potrebuje vzdrževalec

Vzdrževalec potrebuje naslednje datoteke:

- celotna projektna dokumentacija (Projekt_Simulator_Vinsek_Volker.pdf)
- mapo projekta za zagon v Visual Studio 2017 (izvorna koda se nahaja v Simulator.cpp)
- primer tekstovne datoteke *podatki.txt*

D6.3 STRUKTURNI DIAGRAM TER SEZNAM MODULOV IN PODATKOVNIH TOKOV



Vsi moduli se nahajajo v datoteki Simulator.cpp. Pod pojmom moduli so mišljene posamezne funkcije, knjižnice ali pa deli kode v programu.

D6.3.1 Moduli

id	Ime modula	Datoteka	Opomba
0.0	Simulator	Simulator.cpp	celoten program
1.1	normalni_nac	Simulator.cpp	funkcija
1.2	napredni_nac	Simulator.cpp	funkcija
1.3	testni_nac	Simulator.cpp	funkcija
2.1	izpis ukazov	Simulator.cpp	del kode
2.2	pomoc_glavna	Simulator.cpp	funkcija
3.1	branje_dat	Simulator.cpp	funkcija
3.2	pretvorba_temp	Simulator.cpp	del kode
3.3	nakljucno_st	Simulator.cpp	del kode
4.1	iostream	Simulator.cpp	knjižnica
4.2	fstream	Simulator.cpp	knjižnica
4.3	ctime	Simulator.cpp	knjižnica

D6.3.2 Podatkovni tokovi

Oznaka	Ime podatkovnega toka	vrsta (vhod/izhod, kontrola,status)
niz	znakovni niz (izpis na zaslonu)	izhod
vr	številska vrednost	vhod/izhod
nak	naključno število z intervala	izhod

D6.3.3 Modul normalni_nac

Je funkcija, katera zajema delovanje normalnega načina delovanja Simulatorja. Izvede se takrat, kadar ni podanega nobenega zagonskega parametra. Vključuje modul za izpis ukazov ter glavne pomoči, navigacija poteka preko menija. Potrebne podatke pridobi iz datoteke. Predstavlja glavno funkcionalnost programa.

D6.3.4 Modul napredni_nac

Kopija funkcije normalni_nac z dodanimi dodatnimi funkcionalnostmi. Vsebuje več izpisov in izračunov.

D6.3.5 Modul testni_nac

Funkcija, namenjena predvsem testiranju delovanja programa. Omogoča vnos poljubnih podatkov, nato pa vrne pravilen izpis. Zaradi razlike v delovanju od prejšnjih dveh modulov, ima ta funkcija implementiran svoj izpis pomoči. Poleg glavnega menija vsebuje tudi podmeni.

D6.3.6 Modul izpis ukazov

Ne gre za posamezno funkcijo, temveč za dele kode znotraj prej navedenih funkcij. Te vrstice so namenjene predvsem izpisu na zaslon ob upoštevanju podanih vrednosti. V okviru tega modula so zajeti vsi izračuni (pretvorba temperature, izbira naključnih vrednosti) in branje podatkov iz datoteke.

Na koncu simulacije se izpiše statistika, njene vrednosti so izpisane z eno decimalko ali pa brez nje (če gre za celo število).

D6.3.7 Modul pomoc_glavna

Funkcija, katera izpiše pomoč za glavni meni. Poklicana je iz funkcij normalni_nac in napredni_nac (ob izbiri * na glavnem meniju).

D6.3.8 Modul branje_dat

Funkcija branje_dat je namenjena pridobivanju podatkov iz datoteke. Vsebuje tudi izpis pomoči in vnos imena datoteke, katero nato program prebere. Ta modul je potreben v vseh režimih delovanja.

D6.3.9 Modul pretvorba_temp

Znotraj vseh funkcij načinov delovanja je del kode, ki pretvarja različne temperaturne enote v stopinje Celzija. Gre za stavka if in if else, katera preverjata, če je željena temperatura večja od 200 oz. večja od 50. Če je vrednost večja od 200, smatramo, da gre za kelvine, zato tej vrednosti odštejemo 273, da dobimo stopinje Celzija. Če pa je vrednost večja od 50, do vključno 200, vrednosti odštejemo 32 in delimo z 1,8 (pretvorba iz Fahrenheitov v stopinje Celzija).

D6.3.10 Modul naključno_st

Modul naključno_st generira celo naključno število znotraj podanega intervala. Po branju iz datoteke se določijo minimalne ter maksimalne dovoljene vrednosti, nato pa se izbere naključna vrednost v tem razponu. Te vrednosti so shranjene v poljih, katerih dolžina je enaka številu simulacij.

D6.3.11 Modul iostream

Osnovna knjižnica za izpis in vnos v konzolo. Uporabljena je tudi pri branju iz datoteke (v kombinaciji z fstream).

D6.3.12 Modul fstream

Knjižnica za delo z datotekami. Uporabljena je v modulu branje_dat, kjer omogoča odpiranje, branje ter zapiranje datoteke.

D6.3.13 Modul ctime

Knjižnica za delo z datumi in časi. V projektu je uporabljena v kombinaciji z razredom srand, da je generacija naključnih števil bolj učinkovita. V nasprotnem primeru lahko pride do ponavljanja naključnih števil.

D6.4 NAJPOMEMBNEJŠI PARAMETRI IN OPISI PODATKOVNIH STRUKTUR

D6.4.1 Struktura vhodne datoteke

Namen te datoteke je pridobitev željenih ambientalnih lastnosti in intervalov. Je tekstovna datoteka s šestimi vrsticami. Spreminjajo se lahko le vrednosti na koncu vrstic, ter vrednosti v intervalih. Paziti je potrebno na to, da so željene lastnosti podane znotraj intervalov. Primer datoteke:

```
TEMPERATURA: 22
VLAZNOST: 45
OSVETLJENOST: 500
INTERVAL TEMPERATURE: [10,40]
STOPNJA VLAZNOSTI: [30,60]
INTERVAL OSVETLJENOSTI: [10,10000]
```

Enote so določene v sistemskih specifikacijah.

D6.4.2 Parametri pri zagonu programa

Program prepozna dva vhodna parametra. Parameter -t zažene program v testnem režimu, parameter -c pa napredni režim načina delovanja. Ob zagonu programa lahko podamo samo en parameter, nikoli oba, saj se program tako ne bo zagnal. Prav tako pa se ne zažene (samo izpiše opozorilo), če namesto teh dveh parametrov vnesemo karkoli drugega. Ob nepravilnem podanem parametru program izpiše opozorilo o nepravilnem vnosu, nato pa se zaustavi.

Po zagonu se v prvi vrstici za imenom ter verziji programa izpiše tudi način delovanja. To velja za testni in napredni način, pri navadnem tega izpisa ni.

D6.5 NATANČNA IDENTIFIKACIJA UPORABLJENIH ORODIJ IN KNJIŽNIC

- Za pisanje in prevajanje izvorne kode smo uporabili Microsoft Visual Studio 2017. Kodo bo najverjetneje mogoče zagnati tudi z najnovejšo različico tega orodja in ostalimi C++ prevajalniki.
- Za izdelavo dokumentacije je bilo uporabljeno orodje Microsoft Word iz paketa Microsoft Office, s tem orodjem smo tudi pretvorili dokumentacijo v format PDF.
- Za izdelavo grafov smo uporabili spletno orodje draw.io, dosegljivo na spletnem naslovu <https://app.diagrams.net/>
- Za izdelavo vhodne tekstovne datoteke smo uporabili beležnico sistema Windows 10, za to datoteko lahko uporabimo tudi druge urejevalnike besedila.

D6.6 POSTOPEK POTREBEN ZA USTVARJANJE IZVRŠILNE KODE

V Visual Studiu odpremo projekt Simulator.sln, nato pa v zgornji vrstici izberemo meni Build. Po odprtju menija izberemo Build Solution. Počakamo nekaj sekund, nato pa si v spodnjem zavihku Output lahko ogledamo »BuildLog«. Po uspešni izgradnji izvršilne kode, program Simulator.exe najdemo v mapi projekta v podmapi Release ali pa Debug, odvisno od nastavljene konfiguracije.

Primer izpisa v oknu Output:

```
1>----- Build started: Project: Simulator, Configuration: Release Win32 -----
1>Simulator.cpp
1>Generating code
1>All 209 functions were compiled because no usable IPDB/IOBJ from previous compilation was found.
1>Finished generating code
1>Simulator.vcxproj -> C:\Users\primozvolker\Documents\Visual Studio 2017\Projects\Simulator\Release\Simulator.exe
1>Done building project "Simulator.vcxproj".
===== Build: 1 succeeded, 0 failed, 0 skipped =====
```

D6.7 POJMOVNIK

D7 UPORABNIŠKI PRIROČNIK

D7.1 IDENTIFIKACIJA DOKUMENTA, POVEZAVA Z OSTALIMI DOKUMENTI

Dokument	
<u>Uporabniški priročnik</u>	
verzija	1.0
datum	23.1.2021
lokacija	datoteka Projekt_Simulator_Vinsek_Volker.pdf
Odgovorna oseba	
	Deni Vinšek
Reference na drugo dokumentacijo	
	Interni standard CVVS 2-2000
	Naročnikove zahteve 2.0
	Sistemske specifikacije 1.0
	Testni primeri 1.0

Program

SIMULATOR

(verzija 2.0)

Uporabniški priročnik
(verzija 1.0)

zadnja sprememba: 24.1.2021

D7.2 NAMEN

Program Simulator je namenjen simulaciji upravljanja ambientalnih lastnosti v prostoru. Uporabnik mora obvladati osnovno delo s PC računalnikom. Poznati mora tudi osnove ukazne vrstice (ukaz cd).

D7.3 STROJNE IN PROGRAMSKE ZAHTEVE

Program za delovanje potrebuje računalnik zmožen poganjati sistem Windows (10). To je procesor z frekvenco 1Ghz ali več, 1Gb (32-bit) oz. 2Gb (64-bit) pomnilnika, 20Gb shrambe in grafično kartico, ki podpira DirectX 9 ali novejši.

D7.4 NAMESTITEV IN ZAGON PROGRAMA

Program namestimo tako, da vse datoteke iz mape (oziroma arhiva) Simulator prekopiramo na poljubno mesto na disk. Poženemo ga lahko s klikom na ikono programa Simulator.exe (le navaden način). Za zagon v drugih načinih se je potrebno najprej premakniti v imenik, kamor smo program namestili, nakar vpišemo ime programa (Simulator.exe). V primeru opozorila, kliknemo več informacij in »Vseeno zaženi«. Program v normalnem načinu lahko tudi tukaj poženemo tako, da njegovo ime vnesemo brez dodatnih argumentov. S programom delamo interaktivno.

Edina **argumenta ukazne vrstice**, ki ju program pozna, sta:

- -t vključi testni režim delovanja programa. Ta režim delovanja je namenjen preizkuševalcu programa in ne končnemu uporabniku.
- -c vključi napredni način delovanja programa, ki je namenjen tudi končnemu uporabniku. Način izpiše dodatne informacije potrebne za »Covid-19 način«

D7.5 NAVODILO ZA UPORABO

Ko program poženemo, se prikaže glavni meni programa. V tem meniju so na voljo opcije za branje iz datoteke, zagon simulacije, pomoč in izhod iz programa. Možnosti v meniju izbiramo tako, da vpišemo številko funkcije oz. zvezdico za pomoč in pritisnemo Enter.

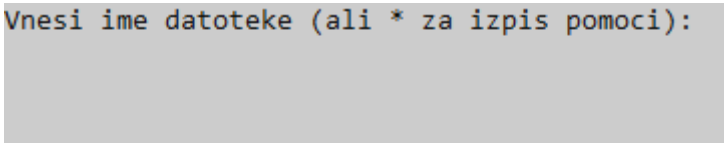
```
Simulator, verzija 2.0 (11.01.2021)
-----
GLAVNI MENI
=====
1) Branje iz datoteke
2) Zagon simulacije
*) Pomoč
0) Izhod iz programa

>>
```

Slika 3 - Zaslón po zagonu programa

Pomoč za uporabo programa je na voljo vedno, ko program čaka na uporabnikov vnos. Prikličemo jo tako, da namesto pričakovanega vnosa, vnesemo zvezdico in pritisnemo enter.

Branje z datoteke



```
Vnesi ime datoteke (ali * za izpis pomoci):
```

Slika 4 - Vnos imena datoteke

Ta funkcija omogoča branje intervalov iz vnaprej pripravljene tekstovne datoteke. Datoteko pripravimo v tekstovnem urejevalniku in jo shranimo kot navadno (.txt) besedilo.

Zgled:

```
TEMPERATURA: 22
VLAZNOST: 45
OSVETLJENOST: 500
INTERVAL TEMPERATURE: [10,40]
STOPNJA VLAZNOSTI: [30,60]
INTERVAL OSVETLJENOSTI: [10,10000]
```

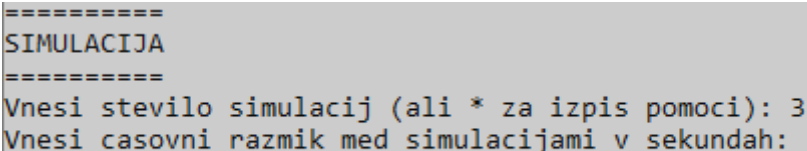
Ko zahtevamo branje polinoma iz datoteke, računalnik čaka na vnos imena datoteke. Tu na običajni način vpišemo ime, če je potrebno pa tudi pot do datoteke.

Nekaj primerov datotek s intervali:

- *C:\Users\Uporabnik\Documents\podatki.txt*
- *intervali.txt*
- *..\podatki.txt*

Zagon simulacije

To funkcijo je možno uporabiti šele, ko že naložimo podatke iz datoteke. Preden se izvede sama simulacija moramo podati 2 parametra.



```
=====
SIMULACIJA
=====
Vnesi stevilo simulacij (ali * za izpis pomoci): 3
Vnesi casovni razmik med simulacijami v sekundah:
```

Slika 5 - Zaslون po izbiri Zagon Simulacije

Prvi parameter je število simulacij, ki jih želimo izvesti. Tu predpisanega limita ni, vnesemo poljubno celo število in pritisnemo Enter. Nato vnesemo še razmik med simulacijami, ki ga moramo podati v sekundah (vrednost je lahko tudi 0) in ponovno potrdimo z Enter. Izvedene simulacije so rezultat naključno izbranih »izmerjenih« parametrov, ki so vedno znotraj intervalov, naloženih iz datoteke. Na koncu prikazanih simulacij so podatki o povprečnih izmerjenih podatkih in njihova odstopanja od povprečja. Za vrnitev na glavni zaslon moramo pritisniti Enter.

```

Simulacija 1:

Izmerjeni parametri
-Temperatura: 22C
-Vlaznost: 48%
-Osvetljenost: 368lx

Zeljeni parametri
-Temperatura: 22C
-Vlaznost: 45%
-Osvetljenost: 500lx

Izvedene operacije:
-Temperatura v prostoru je optimalna
-Izklop vlazilca
-Prizig luci
-----
Simulacija 2:

Izmerjeni parametri
-Temperatura: 35C
-Vlaznost: 55%
-Osvetljenost: 7899lx

Zeljeni parametri
-Temperatura: 22C
-Vlaznost: 45%
-Osvetljenost: 500lx

Izvedene operacije:
-Izklop grelca
-Izklop vlazilca
-Izklop luci
-----
Simulacija 3:

Izmerjeni parametri
-Temperatura: 32C
-Vlaznost: 33%
-Osvetljenost: 6780lx

Zeljeni parametri
-Temperatura: 22C
-Vlaznost: 45%
-Osvetljenost: 500lx

Izvedene operacije:
-Izklop grelca
-Vklop vlazilca
-Izklop luci
-----
Povprecne izmerjene vrednosti:
-Temperatura: 29.6C
-Vlaznost: 45.3%
-Osvetljenost: 5015lx

Povprečna odstopanja od zeljenih vrednosti:
-Temperatura: 7.6C
-Vlaznost: 8.3%
-Osvetljenost: 4603lx

Za vrnitev na glavni meni pritisni Enter.

```

Slika 6 - Rezultati simulacij

D7.6 HIBE PROGRAMA

V drugi verziji programa ni bilo najdenih novih nepravilnosti, vse nepravilnosti iz verzije 1.0 pa so bile odpravljene v verziji 2.0.