|  |
| --- |
| *Avtorja projekta: Primož Volker, Deni Vinšek Verzija: 19. 11. 2020* |
| **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  [ ] D8. Izvorna koda – izpis  [ ] D9. Izvršilna koda  [ ] D10. Opis vseh datotek |
| **Začetek: 15. 10. 2020 Konec: 21. 1. 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:* |

# 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 naslavljajo 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 želenih 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 želenih 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. |
|  |

# Plan projekta

## Identifikacija dokumenta, povezava z drugimi dokumenti

|  |  |
| --- | --- |
| **Dokument: Plan Projekta** | |
| verzija | 1.0 |
| datum | 22. 10. 2020 |
| lokacija | datoteka Projekt\_Simulator\_Vinsek\_Volker.docx |
| **Odgovorni osebi** | Primož Volker, Deni Vinšek |
| **Reference na drugo dokumentacijo** | |
|  | Interni standard CVVS 2-2000 |
|  | Naročnikove zahteve |

## 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.

### 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.

### Omejitve (operacijski sistem, aparaturna 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.

### Rok za zaključitev projekta, skupni stroški

1. 21. 1. 2021
2. Maksimalni skupni stroški izdelave projekta so 2000 EUR.

### 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

### 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

najbolje možno.

### Neizvedljive zahteve

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

### Označevanje verzij

**x.y DDMMLLLL**

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

## Zagotavljanje kakovosti (Načrt preverjanja)

### 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). Kompleten 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 izvorne 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 izvorne 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)

## Naloge in rezultirajoči dokumenti (izbran razvojni model)

**Pregled po produktih in aktivnostih**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Produkt** | **Planirana kompleksnost1** | **Dejanska kompleksnost1** | **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 |  | P. Volker | testiranje + splošni pregled | P. Volker | pisno |  |
| **D4** | **Testni primeri** | 15 testnih primerov |  | D. Vinšek | splošni pregled | D. Vinšek | ustno/pisno |  |
| **D5** | **Testno poročilo** | 5 strani |  | D. Vinšek | splošni pregled | D. Vinšek | pisno |  |
| **D6** | **Načrtovalska dokumentacija** | 2 strani |  | P. Volker | splošni pregled | P. Volker | ustno/pisno |  |
| **D7** | **Uporabniški priročnik** | 2 strani |  | D. Vinšek | splošni pregled | D. Vinšek | pisno |  |
|  | **Program v2.0** | 700 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]](#footnote-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 |  | 6 | 150 |  |  | Deni Vinšek + Primož Volker | Primož Volker |
| **A3** | **Implementacija programa v1.0** | 3. 1. 2021 |  | 20 | 500 |  |  | Primož Volker | Primož Volker |
| **A4** | **Implementacija programa v2.0** | 14. 1. 2021 |  | 10 | 250 |  |  | Primož Volker | Primož Volker |
| **A5** | **Načrtovanje testnih primerov** | 10. 12. 2020 |  | 4 | 100 |  |  | Deni Vinšek | Primož Volker |
| **A6** | **Preverjanje programa v1.0** | 3. 1. 2021 |  | 15 | 375 |  |  | Deni Vinšek | Primož Volker |
| **A7** | **Preverjanje programa v2.0** | 14. 1. 2021 |  | 5 | 125 |  |  | Deni Vinšek | Primož Volker |
| **A8** | **Izdelava kompletne dokumentacije** | 21. 1. 2021 |  | 15 | 375 |  |  | Deni Vinšek + Primož Volker | Primož Volker |
| **A9** | **Prevzem** |  |  | 80 | 2000 |  |  |  | Primož Volker |
|  | **Skupaj napor - stroški** |  |  |  |  |  |  |  |  |

Enota napora: človek-ure

Stroški enote napora: 25 EUR

## Resursi

### Osebje (Kdo bo sodeloval, kakšna je njegova vloga, kakšne morajo biti njegove sposobnosti?)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Oseba** | **AKTIVNOST** | **Vloga** |
| **P1** | direktor podjetja PROstor d. o. o. | * nadzor * prevzem | naročnik |
| **P2** | D. Vinšek | * načrtovanje testnih primerov * testiranje | preverjevalec |
| **P3** | P. Volker | * planiranje projekta * analiza zahtev * načrtovanje * implementacija programa v1.0 * implementacija programa v2.0 * izdelava načrtovalske dokumentacije * prevzem | razvojnik |

### Potrebna programska orodja, knjižnice

|  |  |
| --- | --- |
| **orodje** | **namen, funkcija** |
| Visual Studio 2017 | kodiranje, odpravljanje neustreznosti |
| Microsoft Word | vodenje dokumentacije |
| Winmetr | merilnik kompleksnosti |
|  |  |

### Potrebna strojna oprema

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

## Razdelitev stroškov

Glej točko D2.4.

## Terminski plan projekta

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AKTIVNOST** | **ČASOVNA SKALA** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | **1** | **1** **2** | **2** | **2** **3** | **3** | **3** **4** | **4** | **4** **5** | **5** | **5** **6** | **6** | **6** **7** | **7** | **7** **8** | **8** | **8** **9** | **9** | **9** **10** | **10** | **10** **11** | **11** | **11** **12** | **12** | **12** **13** | **13** | **13** **14** | **14** | **14** **15** | **15** |  |
| 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** | **2** **3** | **3** | **3** **4** | **4** | **4** **5** | **5** | **5** **6** | **6** | **6** **7** | **7** | **7** **8** | **8** | **8** **9** | **9** | **9** **10** | **10** | **10** **11** | **11** | **11** **12** | **12** | **12** **13** | **13** | **13** **14** | **14** | **14** **15** | **15** |  |
| 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

## Pojmovnik

|  |  |
| --- | --- |
| **pojem** | **razlaga** |
| naročnik | PROstor d. o. o. |
|  |  |
|  |  |

## Priloge

### Opisi uporabljenih strategij

1. **Opis strategije: Prisotnost zahtev (Z)**
2. 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.
3. Testirni model je seznam zahtev.
4. **Pravilo za načrtovanje testnih primerov**: Za vsako zahtevo tvori najmanj en testni primer. Vhodne podatke si poljubno izberi.
5. Z načrtovanjem testnih primerov lahko začnemo, ko so zahteve postavljene.
6. Testirna strategija je izčrpana, ko preverimo prisotnost vsake zahteve v seznamu.
7. **Opis strategije za preverjanje robustnosti (R)**
8. Strategija je uporabna je v vseh primerih, kjer je zahtevana robustnost in je možno tvoriti opis vhodne domene.
9. Predpostavka o nepravilnosti: program ni robusten, čeprav bi moral biti. S to strategijo ne bomo odkrili nepravilnosti, ki se pojavljajo pri procesiranju veljavnih podatkov.
10. Testirni model je opis vhodne domene.
11. **Pravilo za načrtovanje testnih primerov**: V vhodni domeni in identificiraj prepovedane razrede. Za vsak prepovedan razred tvori en testni primer.
12. Z načrtovanjem testnih primerov lahko začnemo, ko je opisana vhodna domena.
13. Testirna strategija je izčrpana, ko smo pokrili vse neveljavne razrede v vhodni domeni. Zgornje število testnih primerov je enako številu neveljavnih razredov.
14. **Opis strategije: ugibanje nepravilnosti (U)**
15. Strategija je splošno uporabna.
16. Predpostavlja se, da je prisotna določena nepravilnost ali napaka.
17. Testirni model je seznam potencialnih nepravilnosti oziroma napak.
18. **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.
19. Z načrtovanjem testnih primerov lahko začnemo, ko je imamo pripravljen seznam.
20. Testirna strategija je izčrpana, ko smo pokrili celoten seznam. Zgornje število testnih primerov je enako številu napak oziroma nepravilnosti v seznamu.
21. **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.

# Sistemske specifikacije

## Identifikacija dokumenta, povezava z drugimi dokumenti

|  |  |
| --- | --- |
| **Dokument: Sistemske specifikacije** | |
| verzija | 1.0 |
| datum | 10. 11. 2020 |
| lokacija | datoteka Projekt\_Simulator\_Vinsek\_Volker.docx |
| **Odgovorni osebi** | |
|  | Primož Volker, Deni Vinšek |
| **Reference na drugo dokumentacijo** | |
|  | Interni standard CVVS 2-2000 |
|  | Naročnikove zahteve |
|  |  |

## 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.

## Zahteve glede posameznih karakteristik

Kritične zahteve so označene s klicajem.

### Korektnost oziroma funkcionalnost

Glej točko D 3.7.

### Zanesljivost

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

### Testabilnost

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

### Prenosljivost

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

### 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.

### Razumljivost

Ni kvantitativnih in kvalitativnih zahtev.

### Varnost

Ni kvantitativnih in kvalitativnih zahtev.

### Vzdrževalnost

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

### Zmogljivost

Ni kvantitativnih in kvalitativnih zahtev.

Program se bo preverjal na računalniku z i5 procesorjem 8 generacije in 8Gb pomnilnika. Program bo deloval podobno na vseh sodobnih računalnikih.

## Omejitve in druge zahteve

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

start simulator.exe-t

Kretnica –t požene program v testnem režimu delovanja.

## Opis sistema

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

### 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:
   1. Vnos podatkov v tekstovno datoteko in shranjevanje
   2. Poženemo program Simulator.exe
   3. Program prebere podatke in jih obdela
   4. Glede na vhodne podatke izvede ustrezen ukaz iz pravilnostne tabele
4. Končno stanje: izpisano sporočilo ob zaključku delovanja

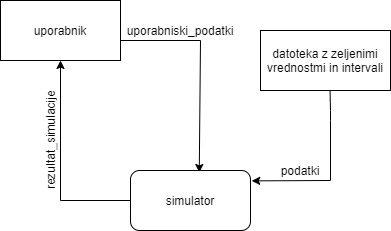
TVZ\_2. Navadni režim

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:
   1. Vnos intervalov v tekstovno datoteko in shranjevanje
   2. Poženemo program Simulator.exe
   3. V začetnem meniju izberemo možnost za zagon simulacije
   4. Program prebere interval in v obsegu sam določi vrednosti podatkov
   5. Glede na vhodne podatke izvede ustrezen ukaz iz pravilnostne tabele
4. Končno stanje: izpisano sporočilo ob zaključku delovanja

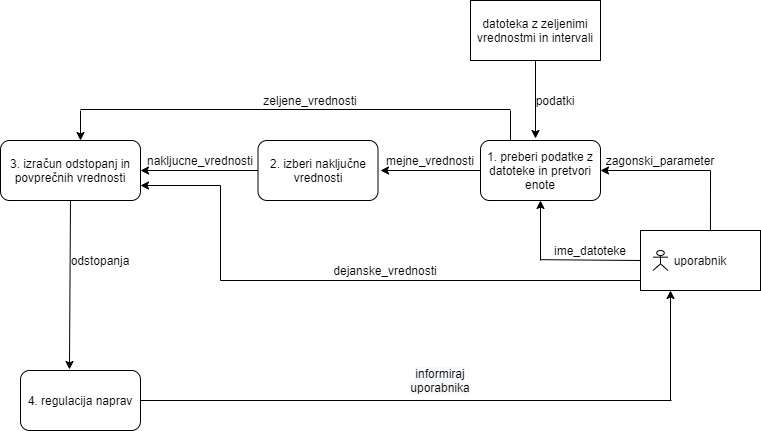
TVZ\_3. Uporabnik zahteva pomoč

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

### Diagrami za opis sistema in podsistemov



Slika 2 – Nivo sistema (kontekstni nivo)

Slika 3 - 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.

1. **Izberi naključne vrednosti**

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

1. **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.

1. **Regulacija naprav**

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

## Opis podatkovnih tokov in terminatorjev

### 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] |
| nakljucne\_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.

### Opis terminatorjev za sliki 2 in 3

|  |  |
| --- | --- |
| **ime terminatorja** | **opis** |
| uporabnik | Uporabnik, ki dela s programom Simulator |
| datoteka z zeljenimi 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: 500 [lx]  INTERVAL TEMPERATURE: [10, 40 ] - enota so °C  STOPNJA VLAZNOSTI: [30,60] – relativna vlažnost v %  INTERVAL OSVETLJENOSTI: [10,10000] – enota so lx |

## Podroben opis in indeksiranje funkcij in drugih zahtev, ki jih je potrebno implementirati

Kritične so označene s klicajem.

F1. !! Branje podatkov iz datoteke

Program mora biti sposoben pridobiti podatke iz datoteke. V datoteki bodo intervali, znotraj katerih si bo izbral naključno vrednost.

F2. Pretvorba podatkov

Program mora znati samodejno prepoznati ter pretvoriti temperaturne enote.

F3. ! 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.

F4. Izpis pomoči

Obstajati mora možnost izpisa pomoči.

F5. Kontrola vhodnih podatkov (robustnost).

Program mora kontrolirati vhodne podatke in zavrniti neveljavne vnose v meniju. Opozoriti mora tudi na nepravilen zapis v datoteki.

F6. Izpis verzije ob zagonu programa

Program mora ob zagonu vedno izpisati svojo verzijo.

Z1. Testni način delovanja.

Program mora podpirati testni režim delovanja, v katerem se izpisujejo dodatne

informacije, vnos poteka preko tipkovnice.

## Zunanji videz

Program je bil pognan v normalnem režimu. Pri testnem načinu bi se vrednosti vpisale direktno v program, kjer se testira delovanje simulatorja.

|  |  |
| --- | --- |
| izgled zaslona | opombe |
| zaslon 1 | glavni meni |
| Simulator, verzija 1.00 (03012021)  --------------------------------------------------  GLAVNI MENI  ===========  1) Zagon simulacije  \*) Pomoc  0) Izhod iz programa | V ukazni vrstici se moramo nahajati v mapi projekta.  start Simulator.exe  Ta ukaz zažene program v navadnem načinu. |
| zaslon 2 |  |
| ==========  SIMULACIJA  ==========  Vnesi stevilo simulacij (ali \* za izpis pomoci): 2  Vnesi časovni razmik med simulacijami v sekundah: 3 | Izbrana je bila prva opcija v glavnem meniju.  Prikaže se besedilo za vnos števila simulacij. Vanj vnesemo koliko simulacij naj program izvede. |
| zaslon 3 |  |
| --------------------POMOC--------------------  Pomoc pri zagonu simulacije  =============================================  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.  Za vrnitev na glavni meni pritisni Enter. | V zaslonu 2 je bil vnesen znak \*.  Izpiše se pomoč. |
| zaslon 4 |  |
| ==========  SIMULACIJA  ==========  Simulacija 1:  Izmerjeni parametri  -Temperatura:15C  -Vlaznost:55%  -Osvetljenost:110lx  Zeljeni parametri  -Temperatura:20C  -Vlaznost:45%  -Osvetljenost:500lx  Izvedene operacije:  -Vklop grelca  -Vklop vlazilca  -Prizig luci  ------------------------------  Simulacija 2:  Izmerjeni parametri  -Temperatura:20C  -Vlaznost:40%  -Osvetljenost:2000lx  Zeljeni parametri  -Temperatura:20C  -Vlaznost:45%  -Osvetljenost:500lx  Izvedene operacije:  -Vklop vlazilca  -Izklop luci  ------------------------------  Povprecne izmerjene vrednosti:  -Temperatura:17,5C  -Vlaznost:47,5%  -Osvetljenost:305lx  Povprecna odstopanja od zeljenih vrednosti:  -Temperatura:2,5C  -Vlaznost:7,5%  -Osvetljenost:945lx  Za vrnitev na glavni meni pritisni Enter. | V zaslonu 2 je bilo vneseno število 2 za število simulacij, razmik med simulacijami pa 3 sekunde. Izvedeta se dve simulaciji. Vmes pretečejo tri sekunde.  Na koncu se izpišejo povprečne izmerjene vrednosti ter povprečna odstopanja od željenih vrednosti. |
| zaslon 5 |  |
| --------------------POMOC--------------------  Pomoc za glavni meni.  =============================================  V tem meniju izbiras med tremi moznostmi:  1) Ce vpises stevilo 1 (in pritisnes enter) se ti odpre zaslon za dolocitev stevila simulacij, ter casovni razmik med njimi.  \*) Pri izbiri z znakom \* se ti odpre ta zaslon za pomoc.  0) Z niclo se izvajanje programa zakljuci.  Za vrnitev na glavni meni pritisni Enter. | V glavnem meniju je bila izbrana pomoč (\*). |

## 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

## Prevzemni kriteriji

1. Program mora biti dokumentiran skladno s standardom CVVS-2/2000.
2. Program mora biti preverjen na najmanj 15 testnih primerih. Naročnik bo pripravil tri svoje testne primere, ki ne smejo pokazati na prisotnost večjih hib.

## Pojmovnik

1. Navedi tudi ustrezno enoto (npr.: enota so ure ali stroški ali ljudje ali kombinacija). [↑](#footnote-ref-1)