

TALLINNA ÜLIKOOL

# Testiplaan

Auditooriumi seadmete juhtimise lahendus

Armin Jaemaa

Martin Sütt

Kert-Jan Ots

# Sisukord

<b>Sissejuhatus.....</b>	<b>3</b>
Eesmärgid:.....	3
Testimise strateegiad:.....	3
<b>Skoop.....</b>	<b>3</b>
Testimisse kaasatud valdkonnad:.....	3
Testimisest välja jäetud valdkonnad:.....	3
<b>Kvaliteedieesmärgid.....</b>	<b>4</b>
Peamised eesmärgid:.....	4
Kõrvaleesmärgid:.....	4
<b>Testimisele lähenemine.....</b>	<b>4</b>
Testjuhtumid:.....	4
Manuaalne testimine:.....	4
Automatiseeritud testimine:.....	4
Negatiivne testimine:.....	5
Agilne testimine:.....	5
<b>Rollid ja vastutused.....</b>	<b>5</b>
<b>Sisenemise ja väljumise kriteeriumid.....</b>	<b>5</b>
Sisenemiskriteeriumid:.....	5
Väljumiskriteeriumid:.....	5
<b>Testimise peatamise ja jätkamise nõuded.....</b>	<b>5</b>
Peatamise kriteeriumid:.....	5
Jätkamise kriteeriumid:.....	6
<b>Testistrateegia.....</b>	<b>6</b>
Vea elutsükkel:.....	6
Vigade prioritseerimine:.....	6
Testitavad tüübid:.....	6
<b>Ressursid ja keskkonnatingimused.....</b>	<b>6</b>
Tööriistad:.....	6
Keskkonnad:.....	6
Seadmed:.....	6
<b>Testimise ajakava.....</b>	<b>7</b>
<b>Kinnitused.....</b>	<b>7</b>
<b>Lühendid ja mõisted.....</b>	<b>7</b>

# Sissejuhatus

Antud testplaan on koostatud auditooriumites olevate seadmete juhtimise veebirakenduse (edaspidi rakendus) testimiseks. Rakendus on mõeldud auditooriumites oleva riistvara juhtimiseks kasutades käsureal käivitatud väliseid programme, mis tagastavad rakendusele väljundkoodi (exit code). Meie projekt ei hõlma endas väliste programmide loomist.

Antud testiplaan kirjeldab rakenduse testimise strateegiat, tagamaks, et rakendus on vastavuses funktsionaalsetele nõuetele. Seoses sellega, et projekt ei hõlma rakenduse väliste programmide arendamist, siis testimine keskendub veebirakenduse osale (GUI, käskude edastamine, tulemuste töötlus).

## Eesmärgid:

- Rakendus töötab vastavalt funktsionaalsetele nõuetele
- Tagab stabiilsuse ja arusaadavuse kasutajale

## Testimise strateegiad:

Antud projekti raames kasutame testimiseks kombineeritud lähenemist, mis hõlmab endas:

1. Manuaalset testimist
2. Automaatset testimist
3. Uuritavad testimised
4. Negatiivne testimine
5. Regressioontestid

# Skoop

## Testimisse kaasatud valdkonnad:

- Kasutajaliidese testimine.
- Käskude edastamine.
- Väljundkoodi töötlemine.
- Nuppude lisamine/eemaldamine/liigutamine administraatori vaates.
- Kujunduse muutmine (värvid, fondid, kujundused).
- Andmebaasist korrektsete andmete kättesaadavus.

## Testimisest välja jäetud valdkonnad:

- Riistvara kontrollieritega suhtlevate välised programmid ja väliste programmide sisemine loogika.
- Riistvara töö

# Kvaliteediesmärgid

## Peamised eesmärgid:

- Veenduda, et kasutajaliides on intuitiivne ja vigadeta.
- Tagada, et kõik nõutud funktsionaalsused on täielikult rakendatud ja töötavad.
- Tagada süsteemi töökindlus erinevates keskkondades (brauserid, seadmed).
- Andmete terviklikkus ja vigade käsitlemine.

## Kõrvaleesmärgid:

- Identifitseerida kõik kriitilised vead enne kasutuselevõttu.
- Dokumenteerida testitulemused ja vigade lahendamise protsess. t

Eesmärk	Testimise meetod	Edu kriteerium
Vigadeta kasutajaliides	Käsitsi + automatiseeritud testid	Kriitilised vead parandatud
Funktsionaalsus	Käsitsi + automatiseeritud testid + negatiivsed testid	Kõik olulised ja kriitilised vead parandatud, süsteem käsitleb vigaseid sisendeid korrektselt
Tagada töökindlus	Käsitsi + automatiseeritud testid + negatiivsed testid	Süsteem töötab stabiilselt ja reageerib korrektsetele/vale sisenditele

# Testimisele lähenemine

## Testjuhtumid:

- Luuakse nõuete põhjal ja kattuvad kõik kasutusjuhud (nt nupu lisamine, ajastamine).

## Manuaalne testimine:

- Kasutajaliidese testimine erinevates brauserites (Chrome, Firefox, Edge) ja seadmetes (iOS, Android, tahvelarvutid).
- Kasutajaliidese stabiilsus ja korrektsus.

## Automatiseeritud testimine:

- Seleniumi abil testitakse kasutajaliidese stabiilsust ja reaktsiooniaega.
- Seleniumi abil testitakse nupu vajutuse korral käivitavat funktsiooni.

## Negatiivne testimine:

- Negatiivsete sisendite testimine (vale sisend, tühjad väärtused, ootamatud käsud)

## Agiilne testimine:

- Iga iteratsiooni lõpus testitakse uusi funktsioone ja tehakse regressioonitestid.

## Rollid ja vastutused

Roll	Vastutusalad
Projektijuht	Kommunikatsioon meeskonna ja kliendiga, ajakava jälgimine.
QA	Testimise planeerimine, vigade prioriteetide määramine, aruannete koostamine. Testjuhtumite kirjutamine, testimine, vigade logimine (Wordis), regressioonid.

## Sisenemise ja väljumise kriteeriumid

### Sisenemiskriteeriumid:

- Testimiskeskond on valmis (brauserid, seadmed, server).
- Nõuded ja disainid on kinnitatud.
- Testandmed on loodud (nt näidiskasutajad, nupumallid).

### Väljumiskriteeriumid:

- Kõik kriitilised ja olulised vead on parandatud.
- 100% nõuete katvus on saavutatud kriitiliste vigade korral.
- Klient on kinnitanud vastuvõtutestide tulemusel.

## Testimise peatamise ja jätkamise nõuded

### Peatamise kriteeriumid:

- Leiti kriitiline viga, mis takistab põhifunktsionaalsuse kasutamist.
- Klient muutis oluliselt nõudeid

## Jätkamise kriteeriumid:

- Vead on parandatud ja kontrollitud.
- Muudetud nõuded on dokumenteeritud ja testitud.

## Testistrateegia

### Vea elutsükkel:

1. Viga logitakse google docs dokumendis.
2. Arendaja parandab vea.
3. QA kontrollib parandust ja märgib vea suletuks.

### Vigade prioritiseerimine:

- Kriitiline (P1)
- Oluline (P2)
- Kosmeetiline/visuaalne (P3)

### Testitavad tüübid:

- Must kast: Funktsionaalsuse testimine ilma sisemist loogikat teadmata.
- Integratsioon: Eesrakenduse käivitav käsuriid, mis käivitaks välise programmi.
- Kasutajakogemus: Liidese intuiitiivsus ja vigade teavitused.
- Negatiivsed testid: Rakenduse tagasiside tühja käsu või andmete puudumise korral
- Regressioonitestid: olemasoleva funktsionaalsuse kontroll pärast muudatusi

## Ressursid ja keskkonnatingimused

### Tööriistad:

- Testjuhtumid: Excel/TestRail.
- Automatiseerimine: Selenium (robot framework).
- Veahaldus: Jira/Word.

### Keskkonnad:

- Brauserid: Chrome, Firefox, Safari, Edge (viimased versioonid).
- lokaalne arenduskeskkond, testserver ( pole kindel kas meil on võimalik kasutada)

### Seadmed:

- Tahvlid: iPad
- Arvutid: Windows 11, Mac OS

## Testimise ajakava

Ülesanne	Algus	Lõpp	Märkused
Testiplaani koostamine	01.05.25	15.05.25	Kõik välise programmiga seotud protsessid ei ole testitavad.
Manuaalne GUI testimine	12.06.25	16.06.25	Testitakse kõik nupu toimingud.
Andmesisestusvormide testimine	16.06.25	18.06.25	Testitakse, kas andmed lisatakse korrektselt andmebaasi.
Andmete kättesaadavus andmebaasist	18.06.25	19.06.25	Testitakse kas andmebaasist tulevad andmed on kätte saadud ja õigesti implementeeritud

## Kinnitused

	Ees- ja perekonnanimi	Allkiri
Projekti juht		
Testi juht		

## Lühendid ja mõisted

Lühend	Selgitus
QA	Kvaliteedikindlustus
UAT	Kasutaja vastuvõtutest
GUI	Graafiline kasutajaliides