RTU Liepājas akadēmijas

Dabas un inženierzinātņu centrs

“Code moodle”

Programmatūras projekta pārvaldības plāns

Versija 1.0

CM

Profesionālās bakalaura studiju programmas

“Informācijas tehnoloģija”

2.kursa studenti

Endijs Dārznieks

Kārlis Lācītis

2025

**Dokumenta autoru un izmaiņu lapa**

Autori

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | V.Uzvārds | Struktūrvienība | Amats | Ieguldījums |
| 1. | Endijs Dārznieks |  |  |  |
| 2. | Kārlis Lācītis |  |  |  |

Saturs

[1.IEVADS 5](#_Toc193738794)

[1.1. Projekta pārskats 5](#_Toc193738795)

[1.2. Projekta nodevumi 5](#_Toc193738796)

[1.3. PPPP evolūcija 6](#_Toc193738797)

[1.4. Saistība ar citiem dokumentiem 6](#_Toc193738798)

[1.5. Definīcijas un akronīmi 6](#_Toc193738799)

[2. PROJEKTA ORGANIZĀCIJA 7](#_Toc193738800)

[2.1. Procesa modelis 7](#_Toc193738801)

[2.2. Organizacionālā struktūra 8](#_Toc193738802)

[2.3. Organizacionālās robežas un saskarnes 8](#_Toc193738803)

[2.4. Projekta atbildīgās personas 9](#_Toc193738804)

[3. PĀRVALDĪBAS PROCESS 10](#_Toc193738805)

[3.1. Pārvaldības mērķi un prioritātes 10](#_Toc193738806)

[3.2. Pieņēmumi, atkarības un ierobežojumi 10](#_Toc193738807)

[3.3. Riska pārvaldība 10](#_Toc193738808)

[3.4. Pārraudzības un vadības mehānismi 11](#_Toc193738809)

[3.5. Personāla komplektēšanas plāns 11](#_Toc193738810)

[4. TEHNISKAIS PROCESS 13](#_Toc193738811)

[4.1. Metodes, rīki un tehnikas 13](#_Toc193738812)

[4.2. Programmatūras dokumentācija 13](#_Toc193738813)

[4.3. Projekta atbalsta funkcijas 14](#_Toc193738814)

[4.3.1. Konfigurācijas pārvaldība 14](#_Toc193738815)

[4.3.1.1. Programmatūras konfigurācijas vienumu identificēšana un nosaukšana 14](#_Toc193738816)

[4.3.1.2. Kontrolētās bibliotēkas 15](#_Toc193738817)

[4.3.1.3. Konfigurācijas vienumu pārvaldība 16](#_Toc193738818)

[4.3.2. Testēšana 18](#_Toc193738819)

[4.3.2.1. Testējamie vienumi 18](#_Toc193738820)

[4.3.2.2. Testēšanas vide 18](#_Toc193738821)

[4.3.2.3. Testēšanas kalendārais plāns 19](#_Toc193738822)

[4.3.2.4. Testēšanas stratēģija 19](#_Toc193738823)

[4.3.2.5. Testēšanas procedūra 19](#_Toc193738824)

[4.3.2.6. Testpiemēru izpildes sekmīguma novērtēšanas kritēriji 19](#_Toc193738825)

[4.3.2.7. Testēšanas nodevumi 20](#_Toc193738826)

[4.3.3. Kvalitātes nodrošināšana 20](#_Toc193738827)

[4.3.3.1. Standarti 20](#_Toc193738828)

[4.3.3.2.Dokumentācija 20](#_Toc193738829)

[4.3.3.3.Apskates 21](#_Toc193738830)

[4.3.3.3.Testi 22](#_Toc193738831)

[4.3.3.4.Koda vadība 22](#_Toc193738832)

[4.3.3.4.Vides vadība 23](#_Toc193738833)

[5. DARBU PAKOTNES, KALENDĀRAIS PLĀNS UN BUDŽETS 24](#_Toc193738834)

[5.1. Darbu pakotnes 24](#_Toc193738835)

[5.2. Atkarības 24](#_Toc193738836)

[5.3. Resursu prasības 25](#_Toc193738837)

[5.4. Budžeta un resursu iedalīšana 26](#_Toc193738838)

[5.5. Kalendārais plāns 27](#_Toc193738839)

[ATSAUCES 28](#_Toc193738840)

[PAPILDKOMPONENTI 28](#_Toc193738841)

[ALFABĒTISKAIS PRIEKŠMETU RĀDĪTĀJS 28](#_Toc193738842)

[PIELIKUMI 28](#_Toc193738843)

# 1.IEVADS

# 1.1. Projekta pārskats

*Šajā PPPP nodaļā jādod kopsavilkums par pro­jekta mērķiem (objectives): par no­doda­ma­jiem pro­duktiem, galvenajām darba akti­vi­tā­tēm, gal­venajiem darba pro­duktiem, sva­rī­gākajām šķir­tnēm, vajadzīga­jiem re­sur­siem, galveno (mas­ter) kal­en­dāro plānu (sched­ule) un budžetu. Pro­jek­ta pārskatam jāapraksta šī un citu projektu sais­tība, ja tāda pastāv. Šo pārskatu nevei­do kā pro­jekta pra­sī­bu oficiālu pār­skaitī­jumu. Šajā PPPP no­da­ļā jāparedz at­sauce uz oficiālu pro­duk­ta pra­sī­bu dokumen­tu.*

uzdevums ir: uzskaita Projekta uzdevumus.

Projekta pasūtītājs (turpmāk – Pasūtītājs) ir RTU Liepājas akadēmijas lektors Dzintars Tompsons.

# 1.2. Projekta nodevumi

1. Dokumentācija
   1. Programmatūras prasību specifikācija (turpmāk – PPS)
   2. Programmatūras projektējuma apraksts (turpmāk – PPA)
   3. Programmatūras projekta pārvaldības plāns (turpmāk – PPPP)
   4. Lietotāja rokasgrāmata (turpmāk – LRG)
   5. Testēšanas žurnāls
2. Programmatūra
   1. Sistēmas priekšpuse
   2. Sistēmas servera puse
3. Datubāze
   1. Shēmas fails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodevums | | Datums | Vieta | Daudzums |
| Dokumentācija | |  |  |  |
|  | PPS | 25.03.2025 | Vietnē GitHub.com | 1 |
|  | PPA | 25.03.2025 | Vietnē GitHub.com | 1 |
|  | PPPP | 25.03.2025 | Vietnē GitHub.com | 1 |
|  | LRG | 06.09.2025 | Vietnē GitHub.com | 3 |
|  | Testēšanas žurnāls | 06.09.2025 | Vietnē GitHub.com | 1 |
| Programmatūra | |  |  |  |
|  | Sistēmas priekšpuse | 09.09.2025 | Vietnē GitHub.com | 1 |
|  | Sistēmas servera puse | 09.09.2025 | Vietnē GitHub.com | 1 |
| Datubāze | |  |  |  |
|  | Shēmas fails | 31.08.2025 | Vietnē GitHub.com | 1 |

**1.tabula Projekta nodevumi**

# 1.3. PPPP evolūcija

1) Nosaka, kas plānu sagatavo, saskaņo, apstiprina:

Plānu sagatavo un uztur projekta vadītājs.

Plānu saskaņo analītiķis apstiprina projekta vadītājs.

Plāns tiek pārskatīts katru reizi, kad tiek veiktas kādas jaunas aktivitātes vai ne retāk kā reizi mēnesī.

2) Nosaka atjaunināšanas kārtību:

Plāna atjaunošanas gadījumā analītiķis par to informē klātienes sanāksmē visus projekta dalībniekus.

3) Nosaka uzglabāšanas kārtību:

Šī Plāna jaunāko apstiprināto laidienu atbildīgais speciālists par Plānu norāda vietu kontrolētajā bibliotēkā katalogā 4.3.1.3. Konfigurācijas vienumu pārvaldība repozitorijāun nosūta pieteikumu (e-pasta vēstuli) dokumenta ievietošanai.

# 1.4. Saistība ar citiem dokumentiem

Standarti:

* LVS 67:1996

Pasūtītāja dokumenti:

* Projekta PPS

# 1.5. Definīcijas un akronīmi

| Jēdziens | Skaidrojumi |
| --- | --- |
| Problēmu ziņojums | Ziņojums par, ekspluatācijā nodotajā Programmatūrā, konstatētu problēmu – kļūdu attiecībā pret noteikto programmatūras funkcionalitāti un citām problēmām. |

**2.tabula Jēdzienu skaidrojumi**

| Saīsinājumi | Skaidrojumi |
| --- | --- |
| DB | Datu bāze |
| PV | Projekta vadītājs |

**3.tabula Saīsinajumu skaidrojumi**

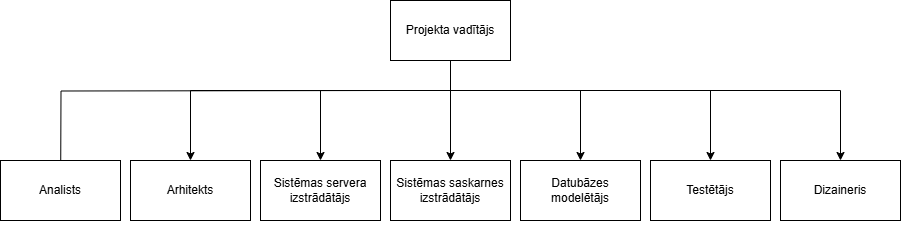
# 2. PROJEKTA ORGANIZĀCIJA

# 2.1. Procesa modelis

| Nr. p.k | Procesa nosaukums | Procesa mērķis | Procesa ieejā | Procesa izejā | Procesa vadītājs |
| --- | --- | --- | --- | --- | --- |
| Projekta procesi |
| **Pārvaldības process** | | | | | |
| 1 | Pārvaldība | Nodrošināt pasākumu kopumu, lai Projekts tiktu plānots un noritētu disciplinētā veidā, nodrošinot plānā paredzēto nodevumu piegādi laikā, ieplānotā budžeta robežās un ievērojot kvalitātes prasības. | Pasūtījums | Pārvaldības plāns | PV |
| **Procesi** | | | | | |
| 2 | Pasūtījuma apstrāde | Vienoties par veicamajiem darbiem, nodevumiem, to izpildes termiņiem un finansējumu | Veicamie darbi | Darbu saraksts,  Gada darba plāns,  Rīcības plāns | PV |
| 3 | Prasību specificēšana | Izvirzīto prasību analīze, specificēšana, dokumentēšana un verificēšana. | Darbu saraksts,  Gada darba plāns,  Rīcības plāns  Pasūtījums | PPS | Analītiķis |
| 4 | Projekta izstrāde | Projekta izstrāde atbilstoši noteiktajām PPS prasībām | PPS | Nodevumi | PV |
| **Atbalstošie procesi** | | | | | |
| ? | Dokumentēšana | Veikt pasūtījumā paredzēto dokumentu sagatavošanu.  Veikt atbilstošo dokumentu sagatavošanu | Pasūtījums, Projekta pārvaldības plāns | Dokuments | Atbildīgais speciālists |
| ? | Konfigurācijas pārvaldība | Nodrošināt konfigurācijas vienumu (dokumentu, programmatūras u.c.) identificēšanu, glabāšanu un izsekojamību, dažādu pārskatu sagatavošanu.  Nodrošināt, lai visas izmaiņas, kuras tiek izdarītas konfigurācijas pārvaldībai pakļautajos vienumos (dokumentācijā vai programmatūrā) tiktu dokumentētas, nodrošinot to izsekojamību. | Konfigurācijas vienums, Izmaiņas | Pārvaldības plāns, izmainīta programma (datu bāzes struktūra), dokumentācija | Atbildīgais speciālists |
| ? | Kvalitātes nodrošināšana | Veikt procesu/produkta kvalitātes nodrošināšanu, lai izstrādātu produktu atbilstoši izvirzītajām prasībām | Mērījuma dati, Produkts | Kārtība, slēdziens, apskates rezultāti | Atbildīgais speciālists |

**4.tabula Procesa modelis**

# 2.2. Organizacionālā struktūra

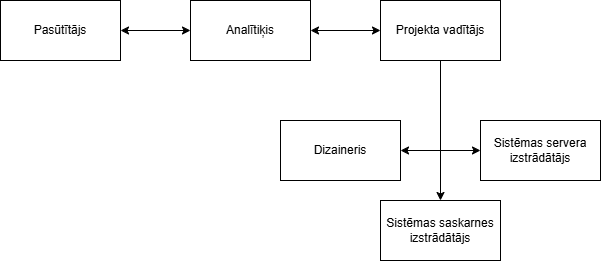
Projekta izstrādi plāno un pārredz projekta vadītājs. Organizacionāla struktūra rādīta 1.attēlā

**1.attēls Organizicionālās struktūras diagramma**

# 2.3. Organizacionālās robežas un saskarnes

Informācijas apmaiņu ar pasūtītāju veic projekta analītiķis. Analītiķis veic informācijas nodošanu projekta vadītājam un vadītājs informāciju izmanto projekta izstrādei. Ja projekta izstrādē ir radušās neparedzētas problēmas, tad projekta vadītājs ziņo analītiķim, kurš veic saziņu ar pasūtītāju.

Organizacionālās saskarnes diagramma rādīta 2.attēlā



**2.attēls Organizacionālās saskarnes diagramma**

# 2.4. Projekta atbildīgās personas

|  |  |  |
| --- | --- | --- |
| **Lomas nosaukums** | **Dalībnieki** | **Lomas darba uzdevumi** |
| Projekta vadītājs | Kārlis Lācītis | Darba organizācija |
| Sistēmas saskarnes izstrādātājs | Kārlis Lācītis, Endijs Dārznieks | Sistēmas klienta puses izstrāde |
| Sistēmas servera izstrādātājs | Kārlis Lācītis, Endijs Dārznieks | Sistēmas API izstrāde |
| Dizaineris | Endijs Dārznieks | Sistēmas klienta puses izstrāde |
| Analītiķis | Endijs Dārznieks | Prasību noteikšana |
| Datubāzes modelētājs | Kārlis Lācītis | Datubāzes modelēšana |
| Testētājs | Kārlis Lācītis, Endijs Dārznieks | Testēšanas žurnāla sagatave, LRG izstrāde |
| Arhitekts | Endijs Dārznieks | Projekta publiskās hostēšanas nodrošināšana |

**5.tabula Projekta atbildīgo personu tabula**

# 3. PĀRVALDĪBAS PROCESS

# 3.1. Pārvaldības mērķi un prioritātes

Projekta pārvaldība un izstrāde tiek organizēta, balstoties uz Agile metodoloģiju. Agile nodrošina elastību un ātru pielāgošanos izmaiņām.

Projekta mērķi:

* Precīza projekta prasību izpilde.
* Funkcionāla un kvalitatīva projekta izstrāde ievērojot noteiktos termiņus.

Projekta prioritātes:

* Produkta funkcionalitāte un kvalitāte.
* Termiņi.
* Efektīva komandas sadarbība.

Ziņošanas mehānismi:

* Iknedēļas sanāksmes – tiek analizēts progress, riski un veicamie darbi.
* Ziņojumi un dokumentācija – svarīgākās izmaiņas tiek paziņotas sanāksmēs un projekta vadības platformā “Trello”.

# 3.2. Pieņēmumi, atkarības un ierobežojumi

Projekta darbība notiek saskaņā ar Darba plānu (DP), Rīcības plānu (RP), Pasūtītāju iesniegtiem Izmaiņu pieprasījumiem (IP) un to papildinājumiem (IPP). Uzturēšana notiek saskaņā ar reģistrētiem kļūdu ziņojumiem (KZ).

# 3.3. Riska pārvaldība

Risku novērtējumu izstrādē veic Projekta vadītājs.

Riski tiek vērtēti katram apstiprinātam darba plāna uzdevumam, tas iekļauj veicamos uzdevumus:

* riska noformulēšana,
* riska novērtēšana,
* stratēģijas izstrādāšana.

Par riskiem un to novērtējumu, analītiķis informē Projekta vadītāju ne retāk kā reizi mēnesī vai pēc Projekta vadītāja ieskatiem. Riski tiek fiksēti IVP protokolos un atskaitēs.

Novērtējot katra darba riskus jāņem vērā sekojoši apstākļi:

* cik efektīva ir sadarbība ar Pasūtītāju. Jo ātrāk un efektīvāk notiek komunikācija ar Pasūtītāju, jo ātrāk un precīzāk tiek sagatavota prasību specifikācija,
* cik pilnīga ir specifikācija un cik ātri ir nodefinētas prasības. Jo ātrāk būs nodefinētas prasības, jo ātrāk programmētāji varēs uzsākt programmēšanu,
* cik sarežģīts ir uzdevums. Jo sarežģītāks, komplicētāks uzdevums, jo lielāka varbūtība, ka uzdevuma realizācijas gaitā var parādīties vēl kādas līdz šim neapzinātas problēmas,
* cik precīzi ir saplānots termiņš. Ir šādu veidu termiņi:
* Ierobežotais termiņš – termiņš, līdz kuram obligāti uzdevumam jābūt izpildītam. Pastāv varbūtība:
* darba spēka pārgrupēšanai,
* uzdevumu prioritātes maiņai,
* citu uzdevumu termiņu nobīdei.
* Plānotais termiņš – termiņš, kuru nosaka vadoties pēc pieredzes un iespējām.
* cik efektīva ir bijusi darba dalīšana, nosakot izpildītāju. Jāņem vērā kāda ir izpildītāja (programmētāja, testētāja):
  + noslodze,
  + kvalifikācija.

# 3.4. Pārraudzības un vadības mehānismi

Atbildīgos par uzdevumiem nosaka PV. Katram Projekta speciālistam PV nosaka individuālos darba uzdevumus, saskaņā ar apstiprināto DP, par kuriem PV informē speciālistu mutiski, pa e-pastu vai Projekta darba sanāksmēs.

Speciālista uzdevumus ievada “Trello” projekta uzdevumu pārvaldības sistēmā PV nozīmēts speciālists. Katram speciālistam ir pienākums darba dienas sākumā piedalīties ikdienas sanāksmē un ziņot PV par darba veikšanai patērētām stundām un komentēt paveikto.

Projekta vadītājs nepieciešamības gadījumā var pieprasīt atsevišķus mutiskus vai rakstiskus ziņojumus par individuālā darba izpildi.

Darbiniekam ir pienākums sekot saviem uzdevumiem un informēt Projekta vadītāju par iespējamajām problēmām individuālā darba plāna izpildē.

Komunikācijas

Projekta sadarbības formas ir “Trello” projekta uzdevumu pārvaldības sistēma, Projekta iekšējās darba sanāksmes un ikdienas sanāksmes. Iekšējās darba sanāksmes tiek organizētas reizi nedēļā. Sanāksmes tiek protokolētas un protokolus uzglabā kontrolētajā bibliotēkā (sk. 4.3.1.3.punkts).

# 3.5. Personāla komplektēšanas plāns

Projektu raksturo vidējas pakāpes sarežģītība, tāpēc nepieciešams nokomplektēt atbilstošas kvalifikācijas komandu. Speciālistu lomas:

* Projekta vadītājs,
* Analītiķis,
* Sistēmas priekšpuses izstrādātājs,
* Sistēmas servera izstrādātājs,
* Sistēmas arhitekts,
* Testētājs

Nepieciešama kvalifikācija:

* Visi speciālisti ir ar vidējo izglītību un vismaz 2-3 gadu pieredzi React projektu izstrādē,
* Kad darbu attīstība izraisa nepieciešamību piesaistīt projektam jaunus izpildītājus, Projekta vadītājs izstrādā konkrētus priekšlikumus par nepieciešamo darbinieku skaitu, kvalifikāciju un atalgojumu un risina šo jautājumu Organizācijas iekšienē noteikto administratīvo jautājumu risināšanas kārtībā.
* Darbinieki tiek piesaistīti uz neierobežotu laiku. Visi savstarpējie jautājumi starp darba devēju un izpildītāju tiek risināti atbilstoši noslēgtajam darba līgumam un Latvijas Republikā spēkā esošai darba likumdošanai.

# 4. TEHNISKAIS PROCESS

# 4.1. Metodes, rīki un tehnikas

Projekta izstrādei ir nepieciešamas nodrošināt speciālistus ar datorsistēmām, kuras spēj nodrošināt ātru koda kompilāciju, kopā ar citiem sistēmas aizmugures procesiem, tādiem kā datubāzes servisiem un servera darbībām. Sistēmu skaits ir jābūt atbilstošs darbinieku skaitam, kuri tieši veic darbu ar veidojamo sistēmu.

Projekta testēšanai ir nepieciešams nodrošināt mobilos tālruņus ar operētājsistēmām Andriod un iOS, tālruņiem ir jāiekrīt vismaz 3 izmēru kategorijās, piemēram, liels tālrunis 6.4 collas, vidējs 6 collas un mazs zem 6 collām

Projekta dokumentācijai tiks izmantots Microsoft Office rīku klāsts, tie ir, Microsoft Word un Microsoft Excel.

Koda kompilācijai, testēšanai kā izstrādes vide tiks izmantots Node.js.

Kā koda redaktors tiks izmantots Visual Studio Code, tas nodrošina lielu klāstu palīgrīku, kas var atvieglot un paātrināt pirmkoda izstrādi.

Datubāzes pārvaldībai tiks izmantots MySQL DBMS, šis rīks nodrošinās vieglu datubāzes izstrādi un vaicājumu testēšanu.

MySQL server tiks izmantots kā datubāzes izstrādes testēšanas vide, gala produktam datubāze tiks uzturēta citā vidē.

Projekta versionēšana tiks veikta ar rīku Git, un informācija tiks glabāta vietnē “GitHub.com”

# 4.2. Programmatūras dokumentācija

*Pro­gram­ma­tūru testēšanas doku­men­tācijas stan­darti:*

ANSI/IEEE Std 829, IEEE Standard for Software Test Documentation

LVS 70:1996, Informācijas tehnoloģija. Programminženierija. Programmatūras testēšanas dokumentācija

*Lieto­tā­ja doku­mentā­ci­jas standarti:*

ANSI/IEEE Std 1063, IEEE Standard for Software User Documentation.

LVS 66:1996, Informācijas tehnoloģija. Programminženierija. Programmatūras lietotāja dokumentācija

Programmatūras izstrādes un uzturēšanas gaitā uzturamie dokumenti ir iedalāmi četrās grupās:

* Projekta darbību reglamentējošie galvenie dokumenti;
* PPS,
* Lietotāju problēmu un kļūdu ziņojumi (P/KZ),
* Pasūtītājam nododamie dokumenti:
* Programmatūras prasību specifikācija (PPS),
* *Programmatūras projektējuma apraksts (PPA),*
* Lietotāja dokumentācija (LRG),
* *Testēšanas dokumentācija (TST)*
* Projekta iekšēji dokumenti:
* Projekta pārvaldības plāns (šis dokuments),

# 4.3. Projekta atbalsta funkcijas

## 4.3.1. Konfigurācijas pārvaldība

### 4.3.1.1. Programmatūras konfigurācijas vienumu identificēšana un nosaukšana

**n. tabula**

Konfigurācijas vienumi

| Pirmkods/  Datu bāze | | Tips | Nosaukums | Piezīmes | |
| --- | --- | --- | --- | --- | --- |
|  | Pirmkods | |  | | |
|  | *MySQL shēmas fails* | *mwb* | *Satur datus par veidojamo datubāzi un ievaddatus* | |
| *Sistēmas klienta un servera puses pirmkods* | *js* |  | |
| *JSON datu tipa fails* | *json* | *Satur aplikāciju pamata informāciju* | |
| *React bibliotēkas saturošs pirmkods* | *jsx* | *Atļauj rakstīt JavaScript ar HTML kopā* | |
| *dokumenti* | *docx, xlsx* | *MS Office dokumenti* | |
| *Html stila fails* | *css* |  | |
| *Servera konfigurācijas fails* | *env* | *Satur datus par autorizāciju un datubāzes savienojumu* | |
| Servera (DB) objekti | |  | | |
|  | *tabulas, datu bāzes saites* | | | |
| *SQL vienības: funkcijas* | | | |
| **Programmatūras dokumentācijā** | | **Tips** | | | **Identifikators** |
|  | | *Programmatūras prasību specifikācija* | | | *CM.PPS.v.* |
| *Programmatūras projektējuma apraksts* | | | *CM.PPA.v.* |
| *Lietotāja rokasgrāmata* | | | *LRG* |
| *Testēšanas dokumentācija* | | | *TST* |
| *Testpiemēri (Test Case, TP)* | | | *TST.TP.nnn.* |
| *Problēmziņojumi (PZ)* | | | *TST.PZ.nnn.* |
| **Pārvaldības**  **dokumentācijā** | | **Tips** | | | **Identifikators** |
|  | | *Programmatūras projekta pārvaldības plāns (PPPP)* | | | *PPPP.v.* |

2.tabula.

Do­ku­men­ta ti­pi

| **Ap­zī­mē­ju­ma** | **Do­ku­men­ta tips** |
| --- | --- |
| Prog­ram­ma­tū­ras do­ku­men­ta tipi: | |
| ARG | Administratora rokasgrāmata |
| LRG | Lietotāja rokasgrāmata |
| PPA | Prog­ram­ma­tū­ras pro­jek­tē­ju­ma ap­raksts |
| PPS | Prog­ram­ma­tū­ras pra­sī­bu speci­fikācija |
| Prog­ram­ma­tū­ras iz­strā­des pro­ce­sa do­ku­men­ta tipi: | |
| KPP | Kon­fi­gu­rā­ci­jas pār­val­dī­bas plāns |
| PPP | Projekta pār­val­dī­bas plāns |
| TST | Tes­tē­ša­nas do­ku­men­tā­ci­ja |
| At­bal­sto­šie do­ku­men­tu tipi: | |
| PLN/PLANS | Plāns |
| PIEM | Pie­mērs |
| PROT | Pro­to­kols |

**6.tabula Konfigurācijas vienumu tabula**

### 4.3.1.2. Kontrolētās bibliotēkas

A black background with white rectangles

AI-generated content may be incorrect.3. attēls. Projekta elektronisko nodevumu mapes struktūra

Projekta kontrolētās bibliotēkas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mape** | | | | **Apraksts** | **Piemērs** |
| ***GGGG\_Projekta\_saisinats\_nosaukums*** | | | | Projekta galvenā mape  *GGGG*- gads | 2025\_CM |
|  | **Individualais\_darbs** | | | Šajā mapē atrodas katra studenta individuālais darbs.  Faila nosaukumu veido šādi: *Vards\_Uzvards\_GGGG\_Projekta\_saisinats\_nosaukums.pdf* | Kārlis\_Lācītis\_2025\_CM.pdf |
|  | **Dokumentacija** | | | Šajā mapē atrodas projekta dokumentācija.  Faila nosaukumus veido šādi:  *GGGG\_Projekta\_saisinats\_nosaukums*\_PPPP.pdf  *GGGG\_Projekta\_saisinats\_nosaukums*\_PPS.pdf  *GGGG\_Projekta\_saisinats\_nosaukums*\_PPA.pdf  *GGGG\_Projekta\_saisinats\_nosaukums*\_LGR.pdf | 2025\_CM\_PPPP.pdf  2025\_CM\_PPS.pdf  2025\_CM\_PPA.pdf  2025\_CM\_LGR.pdf |
|  |  | **Testesana** | | Šajā mapē atrodas testēšanas dokumentācija.  Testpiemēri (TP), Testēšanas žurnāls(TZUR),  Problēmziņojumi(PZ), Problēmziņojumu reģistrācijas žurnāls(PZRZ), Testēšanas kopsavilkums(KOPS).  Akceptestēšanas (AT) dokumentācija: Akceptestēšanas žurnāls (AT\_TZUR), Akceptestēšanas problēmziņojumi (AT\_PZ), Problēmziņojumu reģistrācijas žurnāls(AT\_ PZRZ), Akcepttestēšanas kopsavilkums (AT\_KOPS)  Faila nosaukumus veido saskaņā ar projektā noteikto. | 2025\_CM\_TST\_TP  2025\_CM\_TST\_TZUR  2025\_CM\_TST\_PZ  2025\_CM\_TST\_PZRZ  2025\_CM\_TST\_KOPS  2025\_CM\_TST\_AT\_TZUR  2025\_CM\_TST\_AT\_PZ  2025\_CM\_TST\_AT\_PZRZ  2025\_CM\_TST\_AT\_KOPS |
|  | **Programma** | | |  |  |
|  |  | **Kods** | | Šajā mapē atrodas programmas pirmkoda faili.  Faila nosaukumus veido saskaņā ar projektā noteikto. |  |
|  |  |  | **Klienta\_puse** | Šajā mapē atrodas sistēmas klienta puses pirmkoda faili | App.jsx |
|  |  |  | **Servera\_puse** | Šajā mapē atrodas sistēmas servera puses pirmkoda faili | server.js |
|  |  | **Datu\_baze** | | Šajā mapē atrodas programmas datu bāze.  Faila nosaukumus veido saskaņā ar projektā noteikto. |  |
|  |  | ***Cits*** | | Citas mapes vai faili, kas nepieciešami, lai uzinstalētu programmu.  Faila nosaukumus veido saskaņā ar projektā noteikto. | Readme.docx  Install.docx |

**7.tabula Projekta kontrolētās bibliotekas**

### 4.3.1.3. Konfigurācijas vienumu pārvaldība

Projekta “Code Moodle” izstrādē konfigurācijas pārvaldībai izmanto GitHub, kas ļauj pārvaldīt dažāda tipa informāciju: failu struktūru, failus.

GitHub tiek uzglabātas konfigurācijas vienuma izmaiņas.

GitHub galvenais mērķis ir nodrošināt izmaiņu izsekojamību, uzturot izmaiņu vēsturi.

A group of white containers with black text

AI-generated content may be incorrect.4.attēls. Programmatūras pirmkodu konfigurācijas pārvaldības shēma

Konfigurācijas stāvokļa uzskaite

Pirmkoda vienums var būt šādos stāvokļos (sk. 8.tabula.):

8.tabula. Pirmkodu stāvokļi

| Pirmkoda stāvoklis | | Skaidrojums | Katalogs | Atbildīgais par pirmkodu |
| --- | --- | --- | --- | --- |
| 0 | bāzes versijā | pirmkods netiek mainīts (neatrodas procesā) | Repozitorijā, *production* zarā | Programmētājs |
| 1 | izstrādē | programmētājs veic izmaiņas pirmkoda vienumā | izstrādātāja lokālā darba vide | Programmētājs |
| 2 | iekšējā testā | testētājs veic programmatūras sistēmas testu un ieviešanu | testētāja lokālā darba vide | Testētājs |

Pirmkoda vienuma stāvokļa uzskaiti un kontroli veic testētājs.

## 4.3.2. Testēšana

### 4.3.2.1. Testējamie vienumi

1. Testējamie programmatūras vienumi

* Lietotāju autentifikācija un autorizācija
* Lietotāju reģistrācija
* Uzdevumu un moduļu izveide, rediģēšana un dzēšana
* Studenta uzdevumu izpildes funkcionalitāte
* Vērtēšanas funkcionalitāte
* Komentēšanas sistēma
* Studentu pieņemšana un noraidīšana

1. Netestējamie programmatūras vienumi un cēloņi

* Trešo pušu pakalpojumi, kas tiek izmantoti (jo tos uztur trešās puses)

### 4.3.2.2. Testēšanas vide

1. Nepieciešamā programmatūra
   1. Lokāli
   * Node.js un vajadzīgās pakotnes
   * MySQL DBMS un MySQL server
   * React.js un vajadzīgās pakotnes
   * Moderna Chromium vai Firefox pārlūkprogramma

* Hostēti
  + Moderna Chromium vai Firefox pārlūkprogramma

1. Nepieciešamā aparatūra
   1. Stacionārais vai portatīvais dators ar modernu operētājsistēmu (Windows 10 vai 11, macOS, Linux)
   2. Viedtālrunis
2. Personāls un atbildība
   1. Testētāji: Atbild par testpiemēru izstrādi un izpildi
   2. Izstrādātāji: Atbild par testēšanas rezultātu analīzi un labojumiem
   3. Projekta vadītājs: Uzrauga testēšanas procesu

### 4.3.2.3. Testēšanas kalendārais plāns

Pilnībā visiem testiem tiek izmantots vai nu lokālais vai hostētais risinājums, atšķirībā, kur ir pieejama jaunākā programmatūras versija.

Pilnībā visos testos tiek iesaistīti visi iepriekš minētie personāli.

A screenshot of a computer

AI-generated content may be incorrect. **5. attēls. Testēšanas kalendārais plāns**

### 4.3.2.4. Testēšanas stratēģija

Testpiemēri tiks izstrādāti, balstoties uz lietošanas scenārijiem. Tiks veikti vienības testi mājaslapas saskarnes komponentēm, kā arī manuāla integrācijas testēšana.

### 4.3.2.5. Testēšanas procedūra

1. Izstrādāt testpiemērus atbilstoši funkcionalitātei
2. Izpildīt testus un reģistrēt rezultātus
3. Analizēt rezultātus un izstrādāt labojumus
4. Pārtestēt labotos vienumus

### 4.3.2.6. Testpiemēru izpildes sekmīguma novērtēšanas kritēriji

* Testa izpilde beidzas bez kļūdām
* Lietotāja darbību veikšana bez traucējumiem

### 4.3.2.7. Testēšanas nodevumi

* Testēšanas atskaites (testa gadījumi, veikto testu kopsavilkums)
* Problēmziņojumi un to risinājumi
* Problēmziņojumu reģistrācijas žurnāls
* Testēšanas žurnāls ar izpildītajiem testiem un to rezultātiem

## 4.3.3. Kvalitātes nodrošināšana

### 4.3.3.1. Standarti

Šeit ir saraksts ar projektā izmantotiem standartiem un citiem reglamentējošiem dokumentiem:

• LVS 67:1996 – Programmatūras projekta pārvaldības plāns.

• LVS 70:1996 – Programmatūras testēšanas dokumentācija.

• LVS 66:1996 – Programmatūras lietotāja dokumentācija.

Mainīgo nosaukumiem tika izvēlēta camelCase nosaukumu konvencija un moduļiem PascalCase nosaukumu konvencija. Programmēšanā tika ievērots atomic design princips. Kā komentāru stilu, ievērojām konkrētus, īsus komentārus vietām, kur to ievietošana palīdzētu ar koda paskaidrošanu. Likām komentārus tā lai tie neatkārto kodu. Skatoties uz to ka komanda ir maza, visas koda izmaiņas tika veiktas uz galveno “master” zaru un iesniegtas pa tiešo uz to, netika veidoti atsevišķi zari, priekš specifiskām programmas funkcijām.

### 4.3.3.2.Dokumentācija

**Dokumentācijas veidi projektā**

Projektā “Code Moodle” izmantotā dokumentācija ir iedalāma vairākās grupās:

|  |  |  |
| --- | --- | --- |
| Dokumenta veids | Identifikators | Saturs un mērķis |
| Programmatūras prasību specifikācija | CM.PPS.v | Detalizēti definē visas funkcionālās un nefunkcionālās prasības |
| Programmatūras projektējuma apraksts | CM.PPA.v | Apraksta sistēmas arhitektūru, datu struktūras un komponentes |
| Programmatūras projekta pārvaldības plāns | PPPP.v | Apraksta projekta vadību, plānošanu, riskus, uzdevumus |
| Lietotāja rokasgrāmata | LRG | Nodrošina lietotājiem norādes par sistēmas lietošanu |
| Testēšanas dokumentācija | TST | Satur testēšanas gadījumus, žurnālus, kļūdu aprakstus |
| Problēmziņojumu dokumentācija | TST.PZ.nnn | Reģistrē kļūdas, to izcelsmi, statusu un risinājumus |
| Rīcības un darba plāni | DP, RP | Ietver projekta kalendāro grafiku, uzdevumu sadalījumu un termiņus |

**Dokumentācijas sagatavošanas rīki**

Dokumentācija tiek izstrādāta, izmantojot šādus rīkus:

* Microsoft Word un Excel – formālajiem dokumentiem un tabulām
* GitHub – dokumentu versiju kontrole, koplietošana un publiskā pieejamība

**Dokumentācijas kvalitātes kontrole**

Lai nodrošinātu dokumentu atbilstību kvalitātes prasībām, tiek ievēroti šādi kritēriji un kontroles mehānismi:

* Formāta atbilstība – dokumentiem jāatbilst projekta noteiktajam veidam, faila nosaukumiem un struktūrai (skat. 4.3.1.2.)
* Satura pārskatīšana – pirms publicēšanas dokumenti tiek saskaņoti ar projekta vadītāju un/vai analītiķi
* Atjaunošanas kārtība – dokumentu labojumi tiek veikti tikai pēc saskaņošanas, un tiek glabātas visas iepriekšējās versijas GitHub platformā
* Atbildība – katram dokumentam norādīts autors un atbildīgais speciālists

**Dokumentācijas pārbaude**

Dokumentācijas adekvātumu pārbauda, izmantojot:

* Ekspertu apskates (piemēram, no pasūtītāja vai pasniedzēja)
* Formālas pārbaudes pēc izstrādes un pirms nodošanas
* Salīdzinājumu ar prasībām un testēšanas rezultātiem, lai identificētu neatbilstības

### 4.3.3.3.Apskates

Apskates ir viens no galvenajiem kvalitātes kontroles instrumentiem projektā “Code Moodle”, kas ļauj savlaicīgi identificēt kļūdas, novērtēt atbilstību prasībām un uzlabot gala produkta kvalitāti. Apskates tiek veiktas gan programmatūras, gan dokumentācijas, gan projekta vadības līmenī.

**Apskates veidi un mērķi**

Projektā tiek īstenotas šādas apskates:

Projekta kvalitātes kontroli nodrošina veicot tipveida apskates

| Apskatāmais objekts | Apskates veikšanas veids | Veicējs | Biežums | Apskates dokumentēšanas forma |
| --- | --- | --- | --- | --- |
| Projekta operatīvā norises gaita (tai skaitā risku novērtēšana) | iekšējās darba sanāksmes,  individuāli | Projekta vadītājs, Projekta vadošie speciālisti | Pēc vajadzības, bet ne retāk kā 1 reizi mēnesī | Darba sanāksmes protokols, Projekta atskaites (problēmas un risinājumi) |
| Kalendārais plāns (Darba plāns un/vai Rīcības plāns) | individuāli,  IVP sēdes | Projekta vadītājs | Ne retāk kā 1 reizes mēnesī | Projekta atskaites (problēmas un risinājumi),  IVP sēdes protokols, jaunais mēneša *Rīcības* plāna laidiens.  Atskaites |
| Nododamie dokumenti | individuāli, iekšējās darba sanāksmes | darbinieks, kurš sagatavo dokumentu | Pēc Projekta vadītāja lēmuma dokumenta izstrādes gaitā vai pirms nodošanas | Darba sanāksmes protokols, Projekta atskaites (problēmas un risinājumi)  Atskaites |
| Programmatūras produkta apskate | Individuāli | vadošais programmētājs,  Projekta vadītājs | Pirms produkta nodošanas akcepttestam | e-pasta ziņojumi IVP |
| Testēšanas kļūdu ziņojumu žurnāls | Individuāli | Projekta vadītājs,  vadošais programmētājs | Pirms produkta nodošanas akcepttestam un lietotājam | e-pasta ziņojumi,  ieraksts kļūdu ziņojumu žurnālā un IKZA |
| Jebkura sistēma  Piemēram Moodle |  |  |  |  |

**9.tabula Projekta apsaktes**

**Apskates noslēgšana**

Katra apskate tiek uzskatīta par noslēgtu, ja:

* ir sagatavots un apstiprināts atbilstošs protokols vai komentāru kopsavilkums;
* ir definēti nepieciešamie labojumi;
* labojumu izpilde tiek dokumentēta un pārbaudīta atkārtoti.

Turpmākās darbības

1. Ja apskates laikā tiek konstatēti neatbilstību gadījumi, tiek uzsākta šāda rīcība:
2. Sagatavo detalizētu uzdevumu “Trello” pārvaldības sistēmā.
3. Piešķir atbildīgo par labojumu.
4. Veic atkārtotu apskati pēc izmaiņu izpildes.
5. Atjaunina attiecīgo dokumentāciju vai kodu repozitoriju.

### 4.3.3.3.Testi

**Papildus veicamie testi**

Šie testi tiek veikti, lai pārbaudītu sistēmas vispārējo noturību, drošību un atbilstību lietotāja vajadzībām:

* Drošības testi – autorizācijas, piekļuves kontroles un datu aizsardzības mehānismu pārbaude.
* Savietojamības testi – pārbaude dažādās pārlūkprogrammās (Chrome, Firefox) un uz mobilajām ierīcēm ar Android.

Interfeisa pieejamību uz dažādām ierīcēm.

c) Eksploratīvā testēšana

Šī metode tiek izmantota izstrādes beigu posmā. Testētāji veic manuālu sistēmas pārlūkošanu bez iepriekš definētiem testpiemēriem, lai atklātu:

Neparastus kļūdu gadījumus,

UI/UX nepilnības,

**Testu dokumentēšana**

* Testēšanas žurnālā (TST\_TZUR),
* Problēmziņojumu žurnālā (TST\_PZ, TST\_PZRZ),
* Testēšanas kopsavilkumā (TST\_KOPS).

### 4.3.3.4.Koda vadība

Koda vadība projekta “Code Moodle” ietvaros nodrošina programmatūras pirmkoda versiju uzturēšanu, glabāšanu, aizsardzību un izmaiņu dokumentēšanu visās programmatūras dzīves cikla fāzēs. Šī procesa mērķis ir saglabāt koda integritāti, nodrošināt tā izsekojamību un atvieglot komandas darbu ar vienotu kodu bāzi.

**Versiju kontroles sistēma**

Projekta ietvaros tiek izmantota Git versiju kontroles sistēma. Koda glabāšanai un pārvaldībai tiek izmantots GitHub, kas nodrošina:

* koda vēstures saglabāšanu un izsekojamību,
* iespēju atjaunot iepriekšējās versijas kļūdu gadījumā.

**Koda strukturēšana un organizācija**

Programmatūras pirmkods ir sadalīts loģiskās vienībās:

* Klienta puse – React komponentes
* Servera puse – Express.js servera loģika
* Konfigurācijas faili – .env, package.json.
* Datubāzes shēmas – MySQL .mwb fails

Katra koda komponente tiek glabāta atbilstošā GitHub repozitorija struktūrā (skat. 4.3.1.2. punktu par kontrolētajām bibliotēkām).

**Koda izmaiņu dokumentēšana**

Katrai izmaiņai GitHub repozitorijā jābūt aprakstītai ar:

* skaidru commit ziņojumu, kurā norādīts veiktās izmaiņas būtība,

**Koda stāvokļu uzskaite**

* Koda vienumi tiek pārvaldīti dažādos stāvokļos, kas norādīti konfigurācijas pārvaldības shēmā (skat. 4.3.1.3.):

**Koda aizsardzība**

Piekļuve repozitorijam ir ierobežota tikai komandas dalībniekiem. Papildu drošības mehānismi ietver:

* Paroļu aizsardzību .env failiem, kas netiek augšupielādēti publiski
* Dublēšanas procedūras
* Koda glabāšanu GitHub mākoņserverī, kas nodrošina fizisku un loģisku drošību

### 4.3.3.4.Vides vadība

Vides vadība projekta “Code Moodle” ietvaros ietver metodes un līdzekļus, kas nepieciešami projekta attīstības un uzturēšanas vides identificēšanai, saglabāšanai, aizsardzībai un atjaunošanai visās programmatūras dzīves cikla fāzēs.

**Vides identifikācija un dokumentācija**

Projekta attīstības un testēšanas vides sastāv no šādiem komponentiem:

Izstrādes vide:

* Operētājsistēmas: Windows 10/11, Linux, macOS
* Programmatūras rīki: Visual Studio Code, Node.js, MySQL Workbench, Git
* Koda versiju kontrole: Git + GitHub repozitorijs
* Komandai piešķirtie datorresursi, kas atbilst React projekta izstrādes prasībām

Testēšanas vide:

* Tiek veikta gan lokāli (izmantojot izstrādātāju datorus), gan attālināti (hostētā testēšanas vidē)
* Nepieciešami viedtālruņi vismaz 3 ekrānizmēru kategorijās

Produkcijas vide:

* Hostēta trešās puses serverī (piem., Vercel)
* Piekļuves tiesības un autentifikācija nodrošināta, izmantojot .env konfigurācijas failus

Visas vides tiek dokumentētas projekta konfigurācijas pārvaldības plānā (skat. 4.3.1.3) un tiek periodiski atjaunotas, ja notiek būtiskas izmaiņas infrastruktūrā.

**Fiziskās un loģiskās vides aizsardzība**

Lai aizsargātu programmatūras izstrādes vidi no nesankcionētas piekļuves vai datu zuduma:

* Tiek izmantoti piekļuves ierobežojumi GitHub repozitorijiem, ar lasīšanas/rakstīšanas tiesībām tikai attiecīgajiem komandas dalībniekiem. Repozitorijs ir pieejams publiski vērtēšanai bet to spēj rediģēt tikai projekta komanda.
* .env faili satur sensitīvu informāciju (piem., datubāzes paroles,savionojuma adreses) un netiek iekļauti repozitorijos
* Aizsardzībai pret bojājumiem izmanto versiju kontroles sistēmas (Git), kas nodrošina iespēju atjaunot jebkuru iepriekšējo koda versiju
* Komandu datorsistēmas tiek aizsargātas ar parolēm un atjauninātu pretvīrusu programmatūru

# 5. DARBU PAKOTNES, KALENDĀRAIS PLĀNS UN BUDŽETS

# 5.1. Darbu pakotnes

A screenshot of a computer screen

AI-generated content may be incorrect.Darbu pakotnes un doto darbu pakotņu apraksts rādīts x.attēlā

**6.attēls Darba pakotņu diagramma**

# 5.2. Atkarības

A black background with white rectangles

AI-generated content may be incorrect.Darba pakotņu atkarības ceļa diagramma ir rādītas x.attēlā

**7.attēls Darba pakotņu atkarību ceļa diagramma**

# 5.3. Resursu prasības

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Darba uzdevuma ID | Darba uzdevums | Darbinieki (laiks) | Darbinieku skaits | Programmatūras rīki | Citi resursi |
| 1 | Intervija | Analists (2h) | 2 | MS Word | Diktofons |
| 2 | PPS sagatave | Analists (24h) | 2 | MS Word | Datorsistēma |
| 3 | PPA sagatave | Analists (24h) | 2 | MS Word | Datorsistēma |
| 4 | PPP sagatave | Analists (24h) | 2 | MS Word | Datorsistēma |
| 5 | LRG sagatave | Testētājs (4h) | 1 | MS Word | Datorsistēma |
| 6 | Datubāzes modeļa izstrāde | Datubāzes modelētājs (6h) | 1 | MySQL Workbench | Datorsistēma |
| 7 | Datubāzes sākumdatu izstrāde | Datubāzes modelētājs (4h) | 1 | MySQL Workbench | Datorsistēma |
| 8 | Datu ievade | Sistēmas servera izstrādātājs (12h) | 2 | MySQL Workbench Visual Studio Code MySQL Community Server Node.js Express.js | Datorsistēma |
| 9 | Datu maiņa | Sistēmas servera izstrādātājs (12h) | 2 | MySQL Workbench Visual Studio Code MySQL Community Server Node.js Express.js | Datorsistēma |
| 10 | Datu dzēšana | Sistēmas servera izstrādātājs (8h) | 2 | MySQL Workbench Visual Studio Code MySQL Community Server Node.js Express.js | Datorsistēma |
| 11 | Ielogošanās | Sistēmas servera izstrādātājs (8h) | 1 | MySQL Workbench Visual Studio Code MySQL Community Server Node.js Express.js | Datorsistēma |
| 12 | Reģistrācija | Sistēmas servera izstrādātājs (4h) | 1 | MySQL Workbench Visual Studio Code MySQL Community Server Node.js | Datorsistēma |
| 13 | Sistēmas klienta puses izstrāde | Sistēmas saskarnes izstrādātājs (200h) | 2 | Visual Studio Code Node.js | Datorsistēma |
| 14 | Testēšānas žurnāla sagatave | Testētājs (16h) | 2 | MS Excel | Datorsistēma, mobilās ierīces |
| 15 | Projekta publiskās hostēšanas nodrošināšana | Arhitekts (8h) | 1 |  | Trešās puses hostēšanas risinājums,  Datorsistēma |
| 16 | Dokumentācijas pārskats | Analists (8h) | 2 | MS Word | Datorsistēma |

**10.tabula Projekta nepieciešamie resursi**

# 5.4. Budžeta un resursu iedalīšana

Budžeta un resursu iedalījums redzams x.tabulā.

x.tabulas tabulas norādītais budžets satur tikai projekta atvēlēto sākumbudžetu

|  |  |  |  |
| --- | --- | --- | --- |
| Darba uzdevuma ID | Darba uzdevums | Budžets | Citi resursi |
| 1 | Intervija | 1% |  |
| 2 | PPS sagatave | 2% |  |
| 3 | PPA sagatave | 2% |  |
| 4 | PPP sagatave | 2% |  |
| 5 | LRG sagatave | 1% |  |
| 6 | Datubāzes modeļa izstrāde | 5% |  |
| 7 | Datubāzes sākumdatu izstrāde | 2% |  |
| 8 | Datu ievade | 2% |  |
| 9 | Datu maiņa | 2% |  |
| 10 | Datu dzēšana | 2% |  |
| 11 | Ielogošanās | 4% |  |
| 12 | Reģistrācija | 4% |  |
| 13 | Sistēmas klienta puses izstrāde | 45% |  |
| 14 | Testēšānas žurnāla sagatave | 20% |  |
| 15 | Projekta publiskās hostēšanas nodrošināšana | 5% | Trešās puses hostēšanas risinājums |
| 16 | Dokumentācijas pārskats | 1% |  |

**11.tabula Resursu iedalīšana**

# A close-up of a computer screen AI-generated content may be incorrect.5.5. Kalendārais plāns

**8.attēls Kalendārā plāna tabula**

# ATSAUCES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Dokumenta nosaukums | Identifikators | Autors / Organizācija | Datums |
| 1. | Programmatūras prasību specifikācija | CM.PPS.v | Endijs Dārznieks, Kārlis Lācītis | 25.03.2025 |
| 2. | Programmatūras projektējuma apraksts | CM.PPA.v | Endijs Dārznieks, Kārlis Lācītis | 25.03.2025 |
| 3. | Lietotāja rokasgrāmata | LRG | Endijs Dārznieks, Kārlis Lācītis | 06.09.2025 |
| 4. | Testēšanas dokumentācija | TST | Endijs Dārznieks, Kārlis Lācītis | 06.09.2025 |
| 5. | LVS 67:1996 | – | Latvijas Standarts | – |
| 6. | LVS 70:1996 – Programmatūras testēšanas dokumentācija | – | Latvijas Standarts | – |

# PAPILDKOMPONENTI

Projekta “Code Moodle” PPPP dokumentam ir pievienoti sekojoši papildkomponenti, kas paplašina dokumentācijas detalizāciju un uzlabo lietojamību:

* Akcepttestēšanas dokumentācija (AT\_TZUR, AT\_PZ, AT\_PZRZ, AT\_KOPS)
* Apmācību materiāli (ja nepieciešami pasūtītājam)
* Dokumentācijas šabloni:
  + PPS, PPA, PPPP, LRG veidnes
* Drošības un datu aizsardzības īss plāns

Visi papildkomponenti tiek glabāti projekta kontrolētajās bibliotēkās, atbilstoši norādēm sadaļā 4.3.1.2.

# PIELIKUMI

* Testēšanas žurnāls (TST\_TZUR)
* Testpiemēri (TST\_TP)
* Problēmziņojumu žurnāls (TST\_PZ, TST\_PZRZ)
* Testēšanas kopsavilkums (TST\_KOPS)
* Akcepttestēšanas dokumentācija (AT\_TZUR, AT\_PZ, AT\_PZRZ, AT\_KOPS)
* Dokumentācijas veidnes (.docx, .xlsx formāti)