

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Oliver Puusalu

**Platvormile AUR+ tööluarakenduse ja
integratsioonisüsteemi loomine**

Bakalaureusetöö (9 EAP)

Juhendaja: Jaan Janno

Tartu 2025

Platvormile AUR+ töölaarakenduse ja integratsioonisüsteemi loomine

Lühikokkuvõte:

Tänapäeval on digitaalsete mängude arendamine ja levitamine muutunud oluliselt kättesaadavamaks tänu tehnoloogia arengule ja vabavaraliste tööriistade laiale levikule. Sellest tulenevalt tekkis käesoleva bakalaureusetöö autoril idee luua veebipõhine mänguplatvorm, mis pakub programmeerimishuvilistele võimalust oma mängu esitleda, jagada ning siduda need ühtse kasutajakogemuse ja virtuaalse valuutasüsteemiga. Töö keskendub platvormi kasutajaliidese arendusele ja tehnilistele lahendustele, kirjeldades valitud arendusraamistikke, arhitektuurilisi valikuid ning integreeritud näidismängu. Valminud prototüüp illustreerib loodud süsteemi toimimist ja annab aluse edasiseks arenduseks.

Võtmesõnad:

.NET MAUI, töölaarakenduse loomine, töölaarakenduse disain, mänguplatvorm, kasutajaliides, MVVM, mängude arendus, kasutajakogemus, front-end arendus, platvormi arhitektuur, programmeerimishuvilised

CERCS: P175 Informaatika, süsteemiteooria; T120 Süsteemitehnoloogia, arvutitehnoloogia

Creation of a desktop application and integration system for the AUR+ platform

Abstract:

Nowadays, the development and distribution of digital games has become significantly more accessible due to technological advancements and the widespread availability of open-source tools. As a result, the author of this bachelor's thesis conceived the idea of creating a web-based game platform that allows programming enthusiasts to showcase and share their games while integrating them into a unified user experience and virtual currency system. This thesis focuses on the development of the platform's user interface and technical solutions, detailing the chosen development frameworks, architectural decisions, and the integration of a sample game. The completed prototype illustrates the functionality of the created system and provides a foundation for further development.

Keywords:

.NET MAUI, desktop application development, desktop application design, game platform, user interface, MVVM, game development, user experience, front-end development, platform architecture, programming enthusiasts

CERCS: P175 Informatics, systems theory; T120 Systems engineering, computer technology

Sisukord

Sissejuhatus.....	6
1. Mõisted ja terminid.....	7
2. Taust.....	8
2.1 Töö olulisus.....	8
2.2 Disain.....	9
2.2.1 Kujundus.....	9
2.2.2 Kasutajakesksus.....	10
2.3 Käesoleva töö raames loodud projektile seatud nõuded.....	10
2.4 Kasutatud tarkvara.....	11
2.4.1 .NET MAUI.....	12
2.4.2 Named pipe.....	13
3. Tulemus.....	15
3.1 AUR+ tarkvara.....	15
3.1.1 Sisselogimine.....	15
3.1.2 Registreerimine.....	16
3.1.3 Pealeht.....	17
3.2 Platvormi mündisüsteemi ülesehitus.....	20
3.3 Näidismäng “Gold, Gold, Gold”.....	20
3.3.1 Tehnoloogiline ülesehitus.....	21
3.3.2 Klient-server suhtlus.....	21
3.3.3 Visuaalne ülesehitus ja kasutajaliides.....	22
3.3.4 Näidismängu roll platvormi arenduses.....	22
3.4 Süsteemide integratsioon.....	22
3.5 Nõuete kontroll.....	23
4. Platvormi edasiarendus.....	25
4.1 Allalaaditav mäng.....	25
4.2 Mängude hindamine ja tagasiside süsteem.....	25

4.3 Turvalisuse täiustamine.....	25
4.4 Kasutajakogemuse parandamine Android seadmetes.....	26
Kokkuvõte.....	27
Viidatud kirjandus.....	29
Litsents.....	30

Sissejuhatus

Digitaalsete mängude arendamine ja levitamine on muutunud kättesaadavamaks kui kunagi varem tänu tehnoloogilisele arengule ja vabavaraliste tööriistade levikule. Sellest hoolimata puudub platvorm programmeerijatele, kes on huvitunud kasiinomängudest ning sooviksid oma mängu teistega jagada.

Käesoleva bakalaureusetöö eesmärgiks on välja töötada mänguplatvorm AUR+, mis pakub programmeerimishuvilistele võimalust oma loodud mängu esitleda, jagada ning siduda neid ühise kasutajakogemuse ja virtuaalse valuutasüsteemiga. Platvormi arendamisel lähtuti eesmärgist luua kasutajasõbralik ja laiendatav keskkond, mis toetab mängude üleslaadimist, mängimist ning platvormisest majandust.

Töö käigus keskenduti kasutajaliidese loomisele, mille visuaalid ja funktsionaalsused töötas välja lõputöö autor. Projekti serveriloogika ja andmebaaside ülesehituse arendab teine autor. Töö annab üksikasjaliku ülevaate kasutajaliidese loomise protsessist, alates sisselogimise ja registreerimise vaadetest kuni näidismängu integreerimiseni.

Lisaks praktilisele arendusele käsitletakse töös ka valitud tehnoloogiaid ja arhitektuurilisi otsuseid, sealhulgas MVVM mustri kasutamist MAUI raamistiku kontekstis ning React-põhise näidismängu integreerimist .NET MAUI keskkonda. Tähelepanu pööratakse ka kasutajakogemuse parandamisele visuaalsete lahenduste ja laadimisanimatsioonide kaudu.

Töö lõpptulemuseks on mitmeosaline prototüüp, mis illustreerib platvormi põhifunktsioone ja demonstreerib selle tehnilist toimivust. Töö annab praktilise näite, kuidas luua arendajasõbralik mänguplatvorm, mis ühendab kasutajakontod, mängude halduse ning virtuaalse valuutasüsteemi ühtseks tervikuks.

Järgnevad peatükid kirjeldavad arendusprotsessi ja tulemusi täpsemalt, hõlmates nii kasutajaliidest kui ka platvormi laiendatavuse ja kasutusloogika aspekte.

1. Mõisted ja terminid

Mitmeplatvormiline rakendus (ingl *Multi-platform application*) on rakendus, mis töötab rohkem kui ühte tüüpi seadmel või operatsioonisüsteemil [7].

Named pipe on unikaalne fail, millel on failisüsteemis määratud nimi. Protsessid suhtlevad selle failiga lugemise ja kirjutamise kaudu, töötades sarnaselt tavalisele failile. Eriliseks omaduseks on *named pipe* suutlikkus võimaldada kahepoolset suhtlust. [5]

Protsessidevaheline kommunikatsioon (ingl *inter-process communication*) on mehhanismide kogum, mis võimaldab arvutisüsteemis andmete, teabe või käskluste vahetamist erinevate protsesside vahel. Need mehhanismid on loodud selleks, et lihtsustada protsesside vahelist suhtlust, võimaldada teabe jagamist ning täiustada koordineerimist nende protsesside vahel. [5]

Raamistik (framework) on taaskasutatav tarkvarakomponentide kogum, mis määratleb rakenduse või süsteemi üldise struktuuri. Raamistikud sisaldavad tüüpiliselt abstraktseid klasse ja meetodeid, mida saab spetsiifiliste vajaduste jaoks laiendada ja spetsialiseerida. Need pakuvad arendajale kindlaksmääratud arhitektuurilist alust ja kujundusmustreid, mis lihtsustavad keerukate süsteemide loomist. [2]

Teek (ingl *library*) on infoobjektide kogum, mis on loodud üldiseks korduvaks kasutamiseks¹.

Token (ingl *token*) on objekt, mis tõendab õigust sooritada mingit toimingut².

Kasutajakogemus (ingl *user experience*) on emotsioonid, mida inimene kogeb toote, süsteemi või teenuse kasutamisel. See mõiste sisaldab inimese praktilisi aspekte nagu kasulikkus, kasutusmugavus ja süsteemi efektiivsus. Kasutajakogemus on oma olemuselt subjektiivne näitaja, sest väljendab konkreetse isiku tundeid ja mõtteid teatud süsteemi kohta [8].

¹ <https://akit.cyber.ee/term/4257-teek>

² <https://akit.cyber.ee/term/2213-token-1>

2. Taust

Selles peatükis antakse ülevaade lõputöö teoreetilisest taustast. Täpsemalt selgitatakse töö olulisust ning seatud eesmärged ja nõudeid. Kirjeldatakse kasutatavat tarkvara ning tuuakse välja, kuidas selle valik toetab antud bakalaureusetöö eesmärki.

2.1 Töö olulisus

Käesolev lõputöö on osa projektist, mille viivad läbi autor ja teise aasta informaatikatudeng Madis-Julius Tamberg, mille eesmärk on luua hasartmängude platvorm. Autor lõi platvormile töölaarakenduse, näidiseks platvormile ühe mängu ning kolmesuunalise integreerimise serveri, töölaarakenduse ja mängu vahel. Platvormi serveripoolne lahendus ning üks eraldiseisev mäng luuakse M.-J. Tambergi poolt.

Loodud platvorm on oluline, kuna see loob keskkonna, mis ühendab hasartmängude huvilisi ja programmeerijaid. Platvorm toetab mängude mängimist ning võimaldab arendajatel uusi mänge luua ja levitada.

Platvormile loodud mängu tähtsus on anda teistele programmeerijatele näidis ühest platvormile sobivast mängust. See mäng võtab arvesse artikli *Designing Games with a Purpose: Principles and Practices* [1] tulemusi, mis hõlmavad endas näpunäiteid, kuidas muuta platvorm kasutajasõbralikumaks ja motiveerida mängijaid sagedamini mängima. See artikkel käsitleb mänguarendamise printsiipe, nagu kasutajakesksus, motivatsioon, kaasamine, tasakaalustatud väljakutsed, tagasiside, preemiad, jätkusuutlikkus ja uuendused.

Kuigi peaplatvorm AUR+ ei ole ise mäng, on võimalik neid põhimõtteid edukalt rakendada ka selle kasutajaliidese ja funktsionaalsuse kujundamisel. Näiteks aitab kasutajakesksus ja motiveeriv disain muuta platvormi ligipääsetavamaks ja kasutajasõbralikumaks. Platvormi arendamisel arvestatakse WCAG [9] soovitusetega, et tagada ligipääsetavus erinevate kasutajate vajadusi arvestades. Printsiibid nagu järjepidev tagasiside, mõtestatud navigatsioon ja visuaalne selgus toetavad ühtlasi platvormi üldist kasutatavust ning aitavad kujundada intuitiivset ja kaasavat keskkonda ka mitte-mängulises kontekstis.

Kasutajate üleslaetavate mängude platvormile sobivuse valideerimisel lähtutakse enim, eelmises lõigus nimetatud printsiipidest, kasutajakesksusele ning jätkusuutlikusele. Kasutajakesksusele all mõeldakse, et kuna platvormi fookus on kasiinomängud, siis peab

olema mäng kasutajale aus ning samas arusaadav. Jätkusuutlikkus peaks väljenduma pikaajalise mängukogemuse pakkumises. Nendele kriteeriumitele lisaks peab mäng olema ühilduv loodud platvormiga ja olema suutlik suhtlema AUR+ platvormi serveriga.

2.2 Disain

Kasutajaliidese disaini aluseks võeti „E-teenuste disainimise käsiraamatus“ [8] esitatud põhimõtted, mille kohaselt peab hea e-teenus olema kasutajale intuiitiivne, loogiline ja lihtsalt kasutatav. Disainimisel arvestatakse veebisisu juurdepääsetavuse juhistega (WCAG 2.0) [9], järgides seal toodud soovitusi, mis on antud töös rakendatavad.

2.2.1 Kujundus

Navigatsioonielemendid on paigutatud lehekülje ülaossa, mis vastab üldtunnustatud veebidisaini tavadele ning toetab kasutajate ootusi. Selline paigutus võimaldab ilma eelneva väljaõppeta kasutajatel kiiresti leida põhifunktsioonid ning aitab hoida navigeerimise struktuuri selge ja järjepidevana. Platvorm vastab WCAG 2.0 juhisele 2.4.2 [9], mille kohaselt peab igal lehel olema tuvastatav ja kirjeldav pealkiri. Igal vaatel on määratud pealkiri (nt “AUR+” ja “Login”), mis kajastab lehe sisu ja toetab kasutaja orienteerumist.

Ikoonide asemel on rakenduses valdavalt kasutatud tekstilisi nuppe. Kuigi eksisteerivad standardiseeritud ikoonid, ei ole nende tähendus paljudele kasutajatele alati üheselt mõistetav. Nagu toob välja Aurora Harley artiklis “Icon Usability” [4], võib sama ikoon tähistada erinevates rakendustes erinevaid funktsioone, mis võib viia kasutusvigadeni. WCAG 2.0 juhises 3.3.2 [9] rõhutatakse, et liidese funktsioonid ja eesmärgid peavad olema arusaadavalt esitatud. Sellest tulenevalt on platvormil ikoone kasutatud ainult minimaalsel määral ning nende kõrval on alati ka kirjeldav tekst.

Kogu platvormi värviskeem on valitud nii, et see vastaks WCAG 2.0 kontrastsusnõuetele (juhised 1.4.3 – Minimaalne kontrastsus [9]). Kasutatud on kahte põhimustrit: (a) valge tekst koos musta varjuga heledamatel taustadel ning (b) must tekst kuldsel taustal. Mõlemad lahendused tagavad, et tekst on hästi loetav nii normaalse kui ka vähenenud nägemisvõimega kasutajatele.

2.2.2 Kasutajakesksus

Disainiprotsessi keskmeks oli kasutaja, kelle vajadused, ootused ja käitumismustrid määrasid liidese ülesehituse. Lähtuti “E-teenuste disainimise käsiraamatus“ [8] kirjeldatud põhimõttest, et e-teenuse edukus sõltub otseselt kasutaja tajutud lihtsusest ja sujuvusest. Kõik põhifunktsioonid, nagu mängude sirvimine, info avamine ja üleslaadimine, on paigutatud maksimaalselt ühe kliki kaugusele.

Kasutajakesksust toetab ka see, et süsteem annab igale tegevusele kohest tagasisidet. Näiteks ilmuvad edukate ja ebaõnnestunud toimingute puhul vastavad teavitused, mis vähendavad segadust ja suurendavad kindlustunnet. Kasutajale on kättesaadav tema profiili info, sealhulgas müntide seis ja kontoga seotud andmed, mis on nähtavad igal lehel.

Kogu liidese ülesehitus järgib loogikat, mis võimaldab kasutajal liikuda rakenduses ilma eelneva juhendita. Selleks kasutatakse tuttavaid kujunduselemente, järjepidevat värviskeemi ning selgelt eristatavaid tegevusnuppe. Platvormi disain on üles ehitatud viisil, mis tagab korrektse toimimise ka mobiilseadmetes ja väiksematel ekraanidel, võimaldades ligipääsu kõikidele funktsioonidele sõltumata kasutuskeskkonnast.

Samalaadset lähenemist kasutajakesksuse ja WCAG standardite järgimisel on varem kasutatud Iris Kreinini lõputöös [6], milles rõhutati WCAG suuniste praktilist rakendamist kasutajaliidese kujundamisel ning tajutava lihtsuse saavutamist läbi komponentide paigutuse ja selguse. Käesolevas töös on Kreinini [6] eeskujul samuti lähtutud kasutaja vaatenurgast ning visuaalset disaini kujundatud viisil, mis toetab intuitiivset orienteerumist ja funktsioonide kiiret leidmist.

2.3 Käesoleva töö raames loodud projektile seatud nõuded

Selles alampeatükis on kirjeldatud platvormile esitatavad funktsionaalsed nõuded, lähtudes erinevate kasutajate rollidest, ootustest ning vajadustest. Nõuded on vormistatud “Kasutajana, soovin ..., et ...” struktuuris, mis aitab esile tuua iga funktsionaalsuse taga oleva kasutajakeskse eesmärgi ja motivatsiooni. Selline lähenemine toetab paremini süsteemi disaini kasutajakeskset arendust, aitab tuvastada prioriteetseid funktsioone ning tagab, et lõppkasutaja kogemus jääb arenduse fookusesse. Samuti on kirjeldatud mittefunktsionaalsed nõuded lähtuvalt erinevatest standarditest ja headest tavadest.

Funktsionaalsed nõuded:

1. Kasutajana, soovin kasutada rakendust Windows 11, et saaksin platvormi kasutada ühel kõige levinumal operatsioonisüsteemil.
2. Kasutajana, soovin üles laadida oma mängu, et saaksin neid teistega jagada.
3. Kasutajana, soovin, et platvormil oleks vähemalt üks mäng kohe saadaval, et saaksin rakendust kohe kasutada.
4. Kasutajana, soovin saada oma tegevustele kohest tagasisidet, et saaksin aru, kas minu toiming õnnestus või mitte.
5. Kasutajana, soovin vaadata oma personaalset infot, et saaksin jälgida oma profiili ja tegevusi.
6. Kasutajana, soovin näha igal mängul pilti ja lisainfot, et saaksin enne allalaadimist otsustada, kas mäng mind huvitab.
7. Kasutajana, soovin endale münte juurde lisada, et saaksin platvormi kasutada ka siis, kui ma mündid ära kaotan.
8. Kasutajana, soovin näha pealehel oma müntide koguarvu, et saaksin vajadusel enne mängu minemist münte juurde lisada.
9. Kasutajana soovin, et mängu üleslaadimise nupp oleks platvormi esilehel selgelt nähtav ja hõlpsasti ligipääsetav, et tagada platvormi põhifunktsionaalsuse kohene kergesti leitavus.

Mitiefunktsionaalsed nõuded:

1. Kasutajaliides peab olema arusaadav ja järjepidev.
2. Kasutaja saab visuaalselt kinnitust kõikidele toimingutele.
3. Kõik vaated kohanduvad erinevatele ekraanisuurustele.
4. Tekstide ja tausta kontrast vastab WCAG 2.0 miinimumnõuetele.

2.4 Kasutatud tarkvara

Selles peatükis antakse ülevaade lõputöös kasutatud tarkvaradest, selgitatakse nende valikut ning antakse ülevaade sellest, kuidas need toetavad lõputöös seatud eesmärkide täitmist.

2.4.1 .NET MAUI

.NET MAUI tutvustamisel ning selle tähtsusest kirjutamisel toetuti Microsofti poolt loodud juhendile *Create a cross-platform app with .NET MAUI* [7].

.NET MAUI (*Multi-platform App UI*) võimaldab luua rakendusi erinevatele platvormidele, nagu Windows, Android, iOS ja macOS, kasutades ühtset koodibaasi. See tähendab, et arendajal ei ole vaja kirjutada eraldi koodi igale platvormile, vaid suurem osa rakenduse loogikast ja kasutajaliidesest (UI) saab olla ühes projektis, mis vähendab koodi dubleerimist ja ajakulu. See muudab arenduse kiiremaks ja ressursisäästlikumaks, kuna väheneb vajadus kasutada mitut arendustiimi või spetsialisti erinevate platvormide jaoks.

Ühtse koodibaasi kasutamine tagab ka lihtsama hooldatavuse, sest kui rakendust on vaja uuendada või parandada, saab muudatusi teha ühes kohas, mis kehtivad kõigil platvormidel korraga. See muudab pikaajaliste arenduste hooldamise ja edasi arendamise ökonoomsemaks. .NET MAUI loob võimaluse kliendikogemuse ühtsuseks, kuna kasutajaliides on erinevatel seadmetel visuaalselt sarnane. See tähendab, et olenemata sellest, kas kasutaja avab rakenduse Windowsi arvutis või Androidi telefonis, jääb kasutajaliidese paigutus, kujundus ja kasutusloogika samaks. See suurendab kasutajate rahulolu, kuna nad ei pea erinevatel platvormidel rakendust uuesti õppima ning kasutamine on tuttav.

.NET MAUI toetab andmesidumist ja MVVM (*Model-View-ViewModel*) arhitektuuri, mis võimaldab eraldada rakenduse UI ja äri loogika. See muudab rakenduse koodi struktureeritumaks, lihtsustades selle laiendamist ja testimist. Näiteks, kui on vaja muuta andmete kuvamist, saab seda teha ilma äri loogikat mõjutamata. See omadus on eriti kasulik skaleeritavate ja pika elueaga rakenduste arendamisel. Samuti pakub .NET MAUI tuge platvormispetsiifiliste funktsioonide kasutamiseks. Näiteks võimaldab raamistik rakendusel kasutada seadme sensoreid, kaamerat, GPS-i ja muid riistvaralisi võimalusi ilma, et arendaja peaks kirjutama erinevat koodi igale platvormile. Vajadusel saab .NET MAUI-s siiski kasutada platvormipõhiseid API-sid, et lahendada spetsiifilisi probleeme või kasutada funktsioone, mis on saadaval ainult konkreetsel operatsioonisüsteemil.

.NET MAUI raamistiku kasutamine toetab töö eesmärgi ja tehnilisi nõudeid järgmiselt:

- kuna lõputöö eesmärgiks oli luua rakendus, mis töötab Windowsis ning mis on skaleeritav, pakub .NET MAUI selleks ühtset koodibaasi, mille abil on edasiarendus teistele platvormidele lihtne ning efektiivne;
- .NET MAUI võimaldab kasutada kaasaegseid UI kujundusmalle ning tööriistu, et tagada rakenduse intuiitivsus ja kasutajasõbralikkus. Kasutajaliides on erinevates seadmetes sarnane, mis parandab kliendikogemust.

Madhuri Golde jt. uurimus [3] mitmeplatvormiliste rakenduste arendusraamistike kohta näitab, et sellised rakendused on kuluefektiivsemad ja hooldatavamad kui ühte tüüpi seadmetele või operatsioonisüsteemidele loodud rakendused. Samuti pakuvad need ühtset kasutajakogemust erinevates seadmetes, mis on üks oluline nõue antud töös valminud rakendusele.

Alternatiivsed raamistikud, nagu Flutter ja React Native, on samuti populaarsed, kuid .NET MAUI eeliseks on tugev integreeritus .NET ökosüsteemiga ja võimalus kasutada olemasolevaid C# teeke. Microsofti pikaajaline .NET MAUI toetus on oluline, et platvormi saaks edasi arendada ka peale käesoleva töö valmimist.

2.4.2 *Named pipe*

Käesolevas projektis on protsessidevaheliseks kommunikatsiooniks (IPC) kasutatud meetodit *named pipe*, et korraldada platvormilt allalaetud mängu ja kasutajaliidese vahelist suhtlust. See meetod võimaldab kaht protsessi lokaalselt ühendada ning andmeid vahetada tõhusalt, turvaliselt ning minimaalse viivitusega.

Named pipe on IPC meetod, mis sobib ideaalselt olukordadesse, kus protsessid asuvad samas seadmes ja vajavad stabiilset suhtlust väikese andmemahuga. Antud projektis välistab see vajaduse keeruliste võrguprotokollide järele, tagades süsteemi lihtsuse ja vähendades ressurside tarbimist.

Esmalt pakub *named pipe* meetod kiiret ja madala viitega andmevahetust, mis on oluline reaalajas suhtluse jaoks, näiteks mängu oleku ja kasutaja sisendite edastamisel. *Named pipe* suurendab turvalisust, võimaldades protsessidevahelist suhtlust piirata ainult autoriseeritud protsessidega. Samuti on meetodi lihtne integreerimine märkimisväärne, kuna .NET MAUI platvormi ja .NET-i raamistiku sisseehitatud tööriistad (*System.IO.Pipes*) muudavad *named*

pipe kasutamise arendajale mugavaks, võimaldades sujuvat andmevahetust ilma keerulise seadistamiseta.

H. Karacali, N. Donumi ja E. Cebeli uuringu *Cryptographic Enhancement of Named Pipes for Secure Process Communication* [5] kohaselt on *named pipe* üks kõige tõhusamatest ja turvalisematest lokaalse IPC meetoditest, sobides ideaalselt väikese mahuga ja sagedase suhtluse jaoks. Selle meetodi kasutamise eelised, nagu madal viivitus ja struktureeritud andmevoog, vastavad käesoleva projekti nõuetele.

3. Tulemus

3.1 AUR+ tarkvara

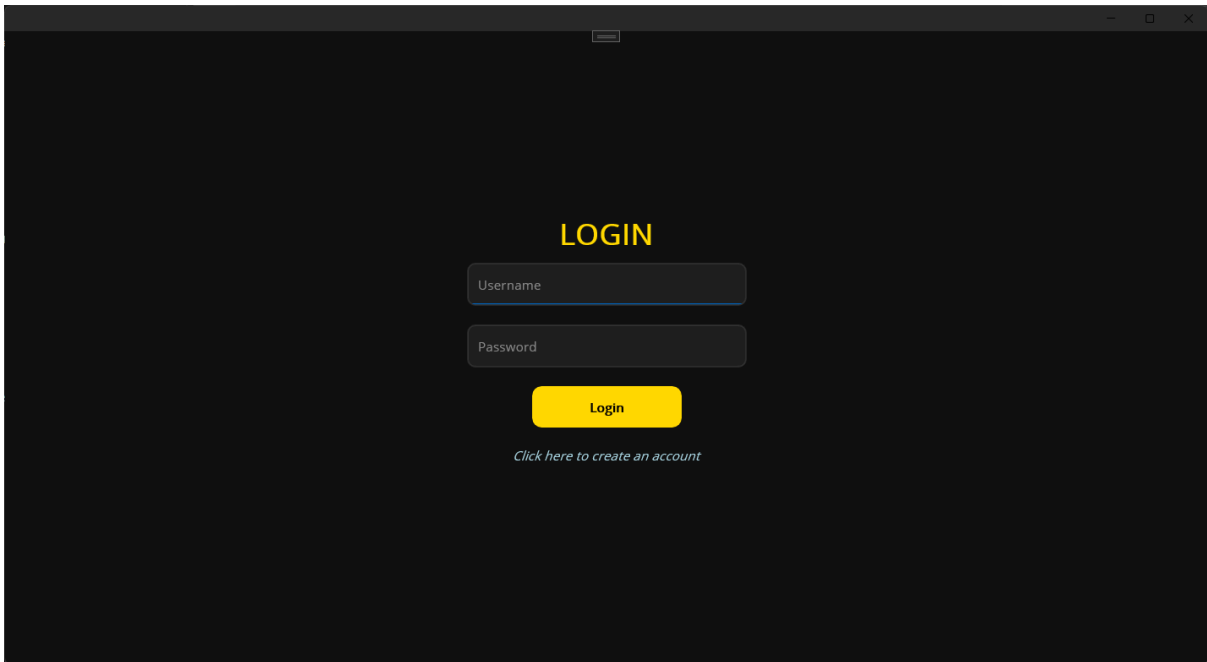
See alampeatükk käsitleb platvormi arendamise tööprotsessi, sealhulgas kasutatud tehnoloogiaid, kujundusotsuseid ning valikuid, mis tehti kasutajaliidese ja funktsionaalsuse arendamisel. Antakse ülevaade arenduse käigus tekkinud probleemidest ja nende lahendustest.

3.1.1 Sisselogimine

Sisselogimisvaade loodi standardse kujunduse järgi. Sellel lehel peab kasutaja sisestama oma unikaalse kasutajanime (*username*) ja parooli (*password*). Sisestatud andmete õigsust kontrollib rakenduse server. Kontrolli tulemuse põhjal toimub üks järgmistest:

- kasutaja logitakse süsteemi sisse;
- kuvatakse teavitus vigasest kasutajanimest või paroolist;
- kuvatakse veateade, kui serveriga ei õnnestu ühendust saada või ilmneb mõni muu tõrge.

Kasutajamugavuse parandamiseks lisati sisestusväljadele (*Entry* komponendid MAUI-s) sisetekstid, mis viitavad, millist teavet tuleb igale väljale sisestada. Mõlemale sisendiväljale on lisatud atribuut `Completed="OnEntryCompleted"`, mis viitab autori loodud funktsioonile. Selle funktsiooni eesmärk on viia kursor automaatselt järgmisele väljale pärast ENTER-klahvi vajutust. Parooli sisestamise järel täidab ENTER-klahv sama funktsiooni nagu nupu "Login" vajutamine. Sisestusväljade asetus on illustreeritud joonisel 1.



Joonis 1. Ekraanivaade sisselogimise lehest.

Kasutajale visuaalse tagasiside andmiseks serveri vastuse ootamise ajal lisati lehele *ActivityIndicator*³ komponent. See kuvatakse poolläbipaistva musta taustaga ning asub ekraani keskel, juhtides kasutaja tähelepanu sellele, et rakendus töötleb parasjagu tema sisestust. Näidik hakkab pöörama automaatselt sisselogimispäringu esitamisel ning kaob, kui serverilt on vastus saadud. Selle eesmärk on parandada kasutajakogemust, andes selge märguande, et süsteem on aktiivne ning kasutajal ei ole vaja mitu korda järjest nuppu vajutada.

3.1.2 Registreerimine

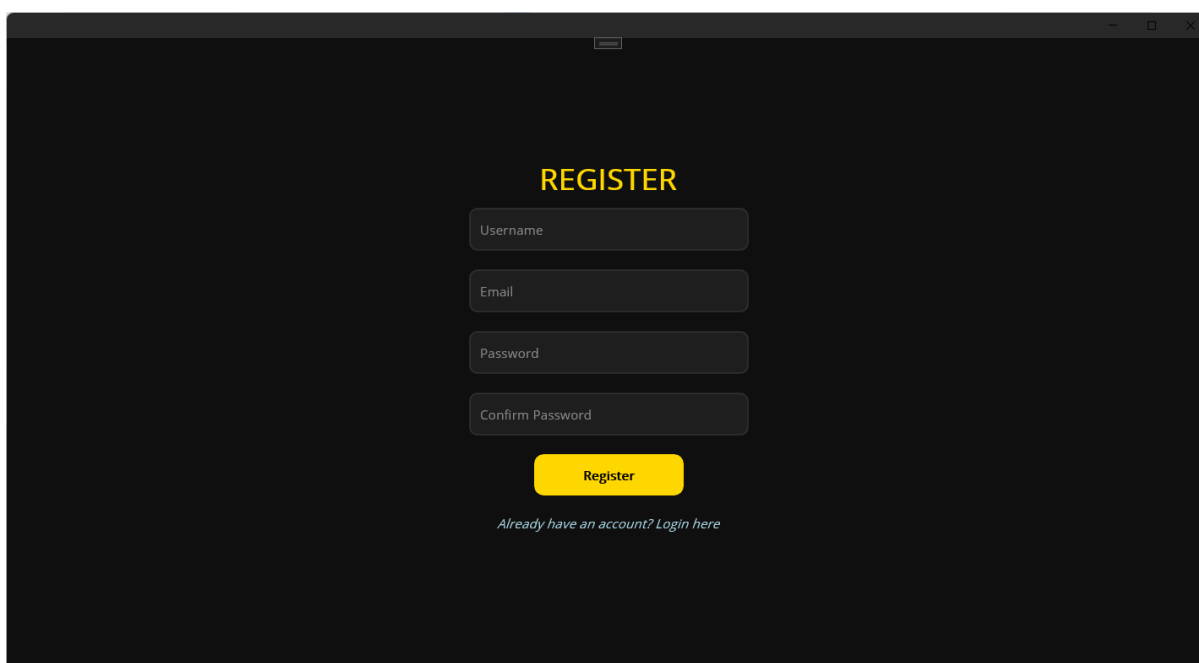
Registreerimisleht järgib sarnast ülesehitust nagu sisselogimisleht ning on samuti loodud standardsete kasutajaliidese praktikate alusel. Kasutajakonto loomiseks peab kasutaja sisestama soovitud kasutajanime (*username*), e-posti aadressi (*email*), parooli (*password*) ning kinnitama parooli uuesti sisestamisega (*confirm password*). Kahekordne parooli sisestamine aitab vältida olukordi, kus kasutaja sisestab parooli ekslikult valesti ning seeläbi ei saa hiljem oma kontole ligi.

Kasutajamugavuse parandamiseks on registreerimisvormi sisestusväljadele lisatud sisetekstid, mis aitavad kasutajal mõista, millist teavet igale väljale sisestada. Samuti on kasutatud funktsionaalsust, mis viib kursori automaatselt järgmisele väljale pärast

³ <https://learn.microsoft.com/en-us/dotnet/maui/user-interface/controls/activityindicator>

Enter-klahvi vajutamist. See on realiseeritud sama mehhanismi abil nagu sisselogimislehel, kasutades parameetrit `Completed="OnEntryCompleted"`, mis viitab vastavale funktsioonile.

Konto loomise protsessi käigus suheldakse serveriga, et kontrollida sisestatud andmete kehtivust ja luua andmebaasis uus kasutaja. Sarnaselt sisselogimisele, kuvatakse siin vastavalt serveri tagasisidele, kas õnnestumis- või veateade. Kuni serverilt vastuse saamiseni kuvatakse kasutajale visuaalne, ootele viitav ning pöörlev laadimisnäidik (*ActivityIndicator*), mis on paigutatud keskele poolläbipaistvale taustale. See aitab kasutajal mõista, et rakendus töötleb sisestatud andmeid ning vältida korduvaid vajutusi. Pärast edukat registreerimist suunatakse kasutaja automaatselt sisselogimise lehele.



Joonis 2. Ekraanivaade registreerimise lehest.

Joonisel 2 on eelnevalt mainitud registreerimislehe visuaalne kujundus.

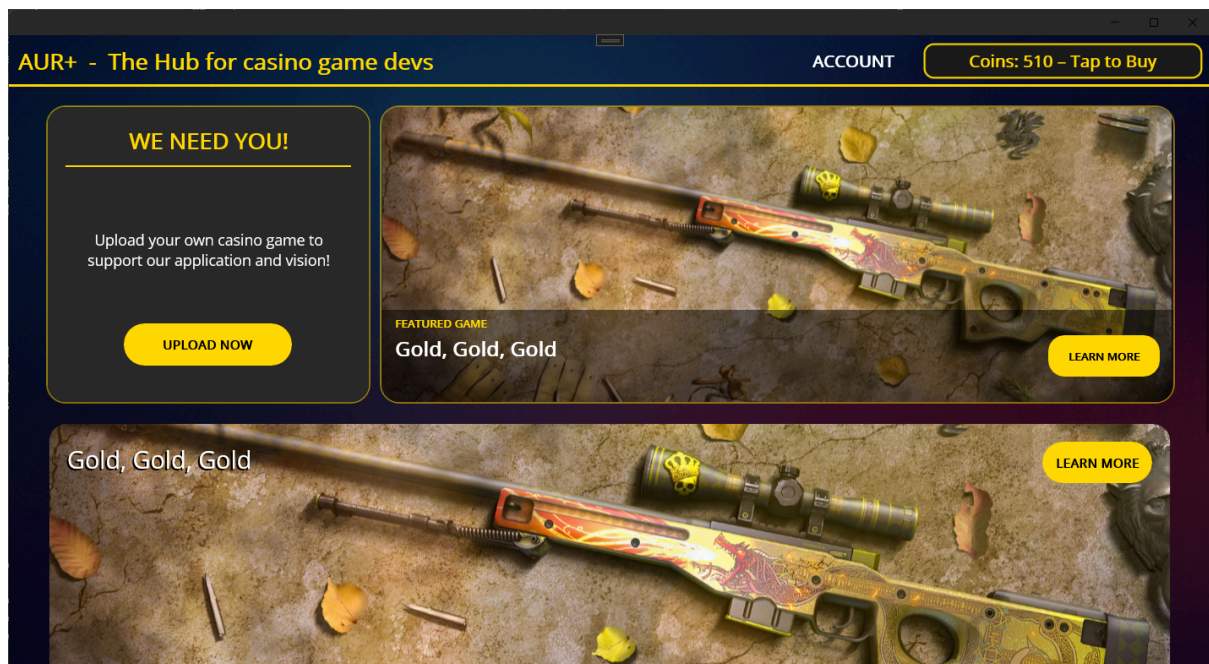
3.1.3 Pealeht

Platvormi avalehe disainimisel lähtuti põhimõttest, et kogu oluline kasutajaga seotud informatsioon oleks maksimaalselt ühe kliki kaugusel. See hõlmab kasutaja identifitseerimisinfot, saadaval olevate mängude loendit ning kasutaja virtuaalsete müntide saldod. Selline lahendus toetab platvormi kasutajasõbralikkust ning võimaldab kiiret orienteerumist, mis on olulise tähtsusega just uutele kasutajatele.

Avalehe keskne komponent on mängude loend, mis kuvatakse dünaamiliselt. Selle kuvamiseks saadab rakendus serverile päringu, mille vastuseks saadakse andmestruktuur mängudest ning nendega seotud metaandmetest. Saadud andmete põhjal luuakse Game-tüüpi objektid, mis salvestatakse rakenduse `Collection<Game> Games` andmestruktuuri. Seejärel renderdatakse iga mängu kohta eraldi visuaalne komponent. Selline lähenemine võimaldab lisada serverisse uusi mänge ilma kliendipoolset koodi muutmata, tagades platvormi laiendatavuse ja hooldatavuse.

Lisaks mänguloendile sisaldab avaleht, kujutatud joonisel 3, ka järgmisi komponente:

- kasutajainfo paneel, kus kuvatakse kasutajanimi, e-posti aadress ja hetkeseis müntidest. Samas vaates saab kasutaja suurendada enda mündisaldot vastava nupu abil;
- esiletõstetud mäng (ingl *featured game*), mis kuvatakse visuaalselt esiletõstetuna ja mille valib süsteem juhuslikult. Selle eesmärk on pakkuda kasutajale soovitusi ning suunata tähelepanu uutele või populaarsetele mängudele;
- mänguinfo vaade, mis avaneb, kui kasutaja klikib mõnele mängule. Avanev vaade sisaldab mängu detailset infot, visuaale ning nuppu mängu mängimiseks (või allalaadimiseks, kui tulevikus lisatakse toetatud mängutüübid);
- mängu üleslaadimise nupp, mis peale edasiarendust suunab kasutaja eraldi lehele, kus on võimalik lisada platvormile enda loodud mäng (bakalaureusetöö esitamise hetkel puudub selle funktsionaalsuse serveripoolne arendus);
- müntide juhtpaneel, mille kaudu saab kasutaja lisada endale virtuaalseid münte, mis toimivad platvormisisese valuutana.



Joonis 3. Ekraanivaade AUR+ pealehest.

Tuleb märkida, et kuna praegusesse platvormi versiooni pole veel lisatud allalaaditavat mängutüüpi, siis funktsionaalsus mängude allalaadimise osas on realiseeritud vaid arhitektuurilisel tasandil ning pole lõplikult testitud. Sellegipoolest on vastavad kasutajaliidese elemendid ette nähtud ning süsteem toetab vastavat laienemist tulevikus.

Lisaks visuaalsetele ja funktsionaalsetele aspektidele on avalehe koodi ülesehitus lahendatud modulaarse arhitektuuriga, mis aitab hoida rakenduse loogika selge ja hooldatavana. Kuna .NET MAUI rakendustes on igal lehel (.xaml) vastav taustafail (.xaml.cs), kuhu on võimalik kirjutada lehe sündmusekäsitlejad ja muu seotud loogika, siis tekib suuremate vaadete puhul oht, et see fail muutub raskesti hallatavaks.

Selle vältimiseks on AUR+ avalehe puhul loodud spetsiaalne vaatemudel nimega "AURplusViewModel.cs", mille ülesanne on hallata andmesidetegevusi serveriga ja eksponeerida kasutajaliidesele vajaminevat andmestikku. Näiteks mängude loetelu, kasutajainfo ja esiletõstetud mäng pärinevad kõik otse "AURplusViewModel.cs" kaudu, võimaldades "AURplus.xaml.cs" failis keskenduda ainult kasutajaliidese käitumise ja nuppude sündmuste käsitlemisele.

Selline lahendus järgib MVVM (*Model-View-ViewModel*) arhitektuurimustrit, mis on soovituslik suurte või laienevate MAUI rakenduste puhul. See võimaldab eristada andmete haldamise ja kasutajaliidese loogika, suurendades koodi taaskasutatavust ja loetavust. See

võimaldab paremat testitavust ning lihtsustab hilisemat arendust, näiteks kui soovitakse muuta seda, kuidas andmeid serverist küsitakse või kuvatakse.

3.2 Platvormi mündisüsteemi ülesehitus

AUR+ platvormil on loodud keskne mündisüsteem, mille eesmärk on toetada kasutajakeskset kogemust, võimaldada lihtsat laiendatavust ning tagada ühtne mängusisene ökosüsteem. Iga registreeritud kasutajaga on seotud unikaalne mündikonto, mille saldod säilitatakse serveripoolselt ning mis on kasutajale nähtav kogu rakenduses.

Platvormi mündid on globaalsed, mis tähendab, et nende väärtus ja kasutusala ulatub üle kõigi platvormil olevate mängude. See võimaldab kasutajal koguda münte ühes mängus ning kasutada neid näiteks teises mängus või uue mängukogemuse avamiseks. Selline ühtne virtuaalvaluuta loob loogilise ja motiveeriva mängutsükli, kus mängudevahelise progressi säilimine suurendab kasutaja seotust platvormiga tervikuna.

Mündisaldo on kasutajale kättesaadav mitmes kohas:

- platvormi avalehel, kus see on nähtav navigeerimisriba paremas servas,
- kasutajakonto vaates,
- igas mängus, kus mündisaldo tuuakse mängu UI-sse mänguserveri kaudu.

Tehniliselt säilitatakse mündisaldo andmeid platvormi peaserveris. Iga kord, kui kasutaja teenib mängus münte või kasutab neid, saadab mänguserver vastava tehingupäringu peaserverile, mis salvestab muudatused andmebaasi. See tagab, et münditehingud on järjepidevad, andmepõhiselt valideeritavad ja säilivad ka peale rakenduse või seadme sulgemist. Tänu sellele saab kasutaja jätkata oma tegevust hiljem samast kohast, sõltumata konkreetsest mängust või seadmest.

Selline arhitektuur toetab hästi platvormi edasiarendamist: näiteks uute mängude lisamine ei nõua muudatusi keskse mündisüsteemi loogikas, kuna iga uus mäng saab kasutada olemasolevat API-liidest, mille kaudu saab kontrollida kasutaja mündiseisu ja teha tehinguid.

3.3 Näidismäng “Gold, Gold, Gold”

Platvormi arendamise käigus loodi demonstratiivne mäng nimega “Gold, Gold, Gold”, mille eesmärk oli näidata AUR+ platvormi võimekust, testida selle tehnilist arhitektuuri ning

pakkuda teistele arendajatele eeskujuks, kuidas luua mäng, mis ühildub platvormi mündisüsteemi ja autentimismudelitega. Käesolevas alampeatükis kirjeldatakse mängu ülesehitust, tehnoloogilisi valikuid, klient-server suhtlust ning rakendatud lahendusi visuaalsete ja loogiliste probleemide lahendamiseks.

3.3.1 Tehnoloogiline ülesehitus

Mäng “Gold, Gold, Gold” loodi veebipõhise kasutajaliidese rakendusena, kasutades React⁴ raamistikku koos JavaScripti⁵ ja Vite’iga⁶. See lahendus võimaldas kiiret arendust ja lihtsat integreerimist platvormi MAUI-põhisesse rakendusse. MAUI rakenduses kasutatakse WebView⁷ komponenti, mille abil renderdatakse veebirakendus otse mobiili- või töölauarakenduse sees.

Kasutaja identifitseerimiseks lisatakse WebView URL-i parameetrina autentimistoken. Mäng saab selle tokeni abil kasutaja tuvastada ja alustada sessiooni. Joonisel 3 on koodinäide WebView komponendi kasutamist .NET MAUI-s.

```
// Find the WebView in the XAML and set its properties
WebView webView = new()
{
    Source = $"http://localhost:4444/?token={token}"
};
```

Joonis 3. Koodinäide WebView komponendi loomisest failis “WebViewRender.xaml.cs”.

See lähenemine võimaldab erinevatel mängudel olla platvormist tehniliselt eraldatud, samal ajal säilitades ligipääsu kesksetele autentimis- ja münditeenustele.

3.3.2 Klient-server suhtlus

Mängu toimimiseks on vajalik kahe-suunaline suhtlus mänguserveri ja AUR+ platvormi peaserveri vahel. Mänguserver valideerib kasutaja tokeni ja haldab mängu käigus tehtavaid münditehinguid. Iga kord, kui mängija võidab auhinna, saadab mänguserver sellekohase info edasi platvormi peaserverile, kes salvestab muudatuse kasutaja kontole. Selline arhitektuur võimaldab säilitada andmete järjepidevuse ja kindlustada turvalise andmetöötluse.

⁴ <https://react.dev/>

⁵ <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

⁶ <https://vite.dev/>

⁷ <https://learn.microsoft.com/en-us/dotnet/maui/user-interface/controls/webview>

Mängu serveripoolne osa loodi Node.js raamistikus, mis sobib hästi REST API loomiseks ja kiireks sündmuspõhiseks andmetöötluks. Kõik tehingud ja päringud toimuvad üle HTTPS-liidese kaudu.

3.3.3 Visuaalne ülesehitus ja kasutajaliides

Mängu kasutajaliides järgib lihtsustatud visuaalset stiili, et keskenduda platvormi toimemehhanismide demonstreerimisele. Kasutajaliideses kasutatud pildid on loodud Microsoft Copilot AI⁸ rakenduse abil. Kuigi visuaalid ei ole lõplikud, sobivad need prototüübiks, mida on võimalik edaspidi kohandada või täiustada.

Visuaalse interaktsiooni osas oli mängu keerukaim arenduslik komponent auhinnaanimatsiooni lõppkoordinaadi määramine. Reacti `<Motion.div>` komponent käivitab animatsiooni kohe pärast renderdamist, mis ei võimaldanud eelnevalt oodata elementide täisrenderdust, et lõppasukohta täpselt määrata.

Probleemi lahendati matemaatilise valemiga, mis arvutas vajaliku lõppasendi, arvestades elementide laiust, vahemaid ning keskpunkti. See valem määras animatsiooni lõppkoha horisontaalsel teljel, lähtudes elementide standardpaigutusest ning keskjoonest.

3.3.4 Näidismängu roll platvormi arenduses

Mäng “Gold, Gold, Gold” teenib kahte olulist eesmärki. Esiteks võimaldab see tehnilise demonstratsioonina testida platvormi erinevaid funktsionaalsusi, sh WebView-põhist integreerimist, autentimisvoogu ning münditehinguid. Teiseks toimib see alusmallina teistele arendajatele, kes soovivad luua oma mängu AUR+ platvormile.

Mäng demonstreerib selgelt, kuidas iseseisev mänguserver saab teha autenditud tehinguid platvormi keskkonnas, säilitades samal ajal loogilise eraldatuse ja turvalisuse. See toetab ka platvormi skaleeritavust, võimaldades uusi mängu lisada ilma peaserveri loogikat muutmata.

3.4 Süsteemide integratsioon

Rakenduse arhitektuur koosneb neljast loogilisest komponendist:

1. mängu kasutajaliides;

⁸ <https://copilot.microsoft.com/>

2. mänguserver;
3. platvormi AUR+ kasutajaliides;
4. platvormi peaserver.

Arhitektuuriline ülesehitus järgib põhimõtet, kus iga mäng suhtleb rangelt vaid omaenda serveriga. Platvormi kasutajaliides ei suhtle mänguserveriga otseselt muudeks tegevusteks peale sessiooni algatamise, mille käigus edastatakse autentimise token.

Kui kasutaja avab AUR+ platvormil mõne mängu, toimub tokeni edastamine mänguserverile kahel viisil, sõltuvalt mängu olemusest. Veebipõhiste mängude puhul saadetakse token URL-i parameetrites mängu protsessile, tööluarakenduste puhul edastatakse see *named pipe*'i kaudu. Mänguserver valideerib saadud tokeni ning pärast edukat kinnitamist saab kasutaja mänguga alustada. Edasine mänguloogika, sealhulgas kõik münditehingud, toimuvad mänguserveri kaudu. Mänguserver vastutab omakorda platvormi peaserveriga suhtlemise eest, et tagada tehingute kehtivus ja järjepidevus kogu süsteemis.

Platvormi kasutajaliides suhtleb peaserveriga ainult juhtudel, kui on vaja lisada kasutajale münte, laadida alla mängude nimekiri või hankida muud kasutajaspetsiifilist infot (nt profiiliandmed, sisselogimine, registreerimine).

Eelnevates alampeatükkides käsitletud autentimisprotsess (sisselogimine ja registreerimine) toimub samuti otse peaserveri vahendusel.

Selleks, et mäng oleks võimeline integreeruma AUR+ platvormi mündisüsteemiga, peab igal mängul olema iseseisev mänguserver, mis suudab suhelda keskse peaserveriga. Selline ülesehitus tagab andmete turvalisuse ja võimaldab hallata mängude autonoomsust platvormi sees.

3.5 Nõuete kontroll

Selles peatükis kirjeldatakse, kuidas lõputöö käigus arendatud AUR+ platvorm vastab varasemalt seatud funktsionaalsetele nõuetele, mis on esitatud alampeatükis 2.3. Eesmärk on hinnata, kas iga nõue on täidetud, osaliselt täidetud või vajab edasist arendamist. Kontrollimise aluseks on rakenduse reaalne käitumine, kasutajaliidese funktsionaalsus ning olemasolevad tehnilised lahendused.

Tabel 1. Platvormile seatud funktsionaalsete nõuete täitmine

Funktsionaalse nõue	Staatust	Märkus
Rakendus on kasutatav Windows 11 seadmetes.	Täidetud	
Kasutaja saab mängu üleslaadida.	Täidetud osaliselt	On loodud sektsioon pealehel mängu üleslaadimiseks, kuid selle serveripoolne arendus on puudulik.
Platvormil on vähemalt üks mängitav mäng	Täidetud	
Kasutaja tegevustele antakse kohene tagasiside.	Täidetud	
Kasutaja saab vaadata oma personaalset informatsiooni.	Täidetud	
Igal mängul on seda kirjeldav infoleht.	Täidetud	
Kasutajal on võimalik endale münte juurde lisada.	Täidetud	
Pealehel peab olema nähtav müntide koguarv.	Täidetud	
Üleslaadimise nupp on hõlpsasti leitav avalehelt.	Täidetud	

Tabelis 1 on esitatud kõigi funktsionaalsete nõuete täitmise staatus, samuti asjakohased kommentaarid, kui funktsionaalsus vajab parandusi või on rakendatud piiratud kujul. Märkustes on kirjeldatud, millises ulatuses nõue on realiseeritud ning kas see vajab edasist tähelepanu enne avalikustamist.

4. Platvormi edasiarendus

Selles peatükis käsitletakse platvormi funktsionaalsusi ja tehnilisi aspekte, mida oleks vaja edasi arendada enne, kui platvormi saab laiemalt kasutusele võtta. Kuigi prototüüp demonstreerib põhilisi funktsionaalsusi, sealhulgas kasutajakontode loomist, mängude lisamist, virtuaalse valuutaga seotud mehhanisme ja näidismängu integreerimist, on reaalseks kasutuseks vajalik mitmete lisavõimaluste lisamine ning töökindluse tagamine.

4.1 Allalaaditav mäng

Kuigi prototüübis on integreeritud üks veebipõhine näidismäng, ei ole veel toetatud täismahus allalaetavate mängude kasutamine. Selleks, et hinnata platvormi võimekust käidelda suuremahulisi ja lokaalselt töötavaid mängu, tuleks luua vähemalt üks mäng, mida kasutaja saab alla laadida ning platvormi kaudu käivitada. Selleks tuleb:

- luua mängude üleslaadimise süsteem, mis toetab failide haldust, metaandmete lisamist ja turvakontrolli;
- integreerida mängukäivitaja, mis võimaldab kasutajal allalaaditud mängu lihtsalt avada ja mängima hakata ning mis suudab kasutaja arvutis avada mängu vastavalt operatsioonisüsteemile;
- jälgida mängu allalaadimisi ja seostada need kasutajakontoga.

Allalaaditava mängu lisamine oleks oluline samm platvormi funktsionaalsuse avardamisel ja pakuks tuge nii veebipõhiste kui ka eraldiseisvatele mängudele.

4.2 Mängude hindamine ja tagasiside süsteem

Üks oluline aspekt kasutajate kaasamisel ja mängude kvaliteedi hindamisel on võimalus anda tagasisidet. Tulevikus võiks lisada hinnangusüsteemi, kus mängijad saavad anda mängudele tärne või hindeid ning lisada kommentaare, et edendada kogukonna kujunemist ja toetada mänguarendajaid.

4.3 Turvalisuse täiustamine

Kuna platvormi kasutajad saavad üles laadida mängu ja hallata virtuaalset valuutat, tuleb edasises arenduses pöörata suuremat tähelepanu andmeturbele. Rakendus tuleb viia

vastavusse GDPR-i nõuetega ning tagada kasutajaandmete kaitse kogu nende elutsükli jooksul. Tuleb kehtestada selged andmetöötluspõhimõtted, sealhulgas kasutaja nõusoleku küsimine andmete kogumiseks ja töötlemiseks, võimalus andmeid eksportida ning soovi korral konto ja sellega seotud info täielikult kustutada.

Lisaks tuleb rakendada autentimismeetmete täiendusi, näiteks kahefaktoriline autentimine (2FA), ning piirata ligipääsu tundlikule teabele vastavalt kasutajaõigustele. Samuti tuleb pöörata tähelepanu serveripoolsete päringute turvalisusele, näiteks sessiooni aegumisele, CSRF-kaitsele ja krüpteeritud andmesidele HTTPS-protokolli kaudu.

4.4 Kasutajakogemuse parandamine Android seadmetes

Kuigi platvorm hetkeseisus on võimalik käivitada Android seadmetes, keskendub prototüüp pigem töölaua kasutajakogemusele. Edaspidi tuleks arvesse võtta rohkem ka mobiilikasutajaid. Platvormi liidest ja mängude käivitamist peaks kohandama väiksematele ekraanidele, samuti võiks arendada natiivseid mobiilirakendusi, mis võimaldaksid paremat jõudlust ja ligipääsetavust.

Kokkuvõte

Bakalaureusetöö raames arendati mänguplatvormi AUR+, mille eesmärk oli pakkuda programmeerimishuvilistele keskkonda, kus nad saavad luua, üleslaadida ja jagada oma arendatud mängu ning kasutada ühise kasutajakogemusega seotud virtuaalse valuutasüsteemi võimalusi. Töö keskmes oli platvormi töölaarakenduse kujundamine ja realiseerimine, mille eest vastutas töö autor, samas kaasautor tegeles serveripoolse loogika ja andmebaasihaldusega. Tegemist on kahe autori koostööprojektiga, mille käigus valmib mitmeosaline funktsionaalne prototüüp.

Platvormi arendus lähtus vajadusest pakkuda väiksematele või iseseisvatele mänguarendajatele kesket ja ligipääsetavat kanalit oma projektide esitlemiseks ja arendamiseks. Sellistel arendajatel jääb tihti puudu olemasolevate suurte mänguplatvormide nähtavusest või võimalustest, mistõttu AUR+ eesmärk on pakkuda just sellele sihtrühmale mõeldud lahendust. Platvormi põhifunktsioonid hõlmasid kasutajakontode loomist ja haldamist, mängude lisamist ja sirvimist, ning platvormisisese valuuta kasutamist, mille kaudu saab motiveerida kasutajaid mängude loomisel ja nende proovimisel.

Töös kirjeldatakse põhjalikult kasutajaliidese arendust, sealhulgas MVVM arhitektuurimustri rakendamist .NET MAUI raamistikus. Samuti käsitleti React-põhise näidismängu integreerimist MAUI keskkonda WebView komponendi kaudu, mis võimaldas ühenduse luua erinevate tehnoloogiate vahel. Tähelepanu pöörati ka visuaalsete lahenduste väljatöötamisele, näiteks laadimisanimatsioonidele ja intuitiivsele kasutajavoole, mille eesmärk oli tagada sujuv ja esteetiliselt meeldiv kasutuskogemus.

Kuna töö autori vastutusvaldkonnaks oli kasutajaliides ja kliendipoolne loogika, siis keskenduti töös järgnevatele detailidele: ekraanivaadete ülesehitus, navigatsioon, andmestruktuuride sidumine kasutajaliidesega ning kasutuskogemuse üldine optimeerimine. Arenduses lähtuti praktilisusest ja laiendatavusest, et loodud platvormi oleks võimalik hiljem edasi arendada uute funktsioonide ja mängudega.

Töö tulemusel valmis prototüüp, mis demonstreerib platvormi peamisi funktsionaalsusi ja pakub aluse edasisteks arendusetappideks. Valminud süsteem annab praktilise näite, kuidas luua arendajasõbralik ja kasutajale mugav mängukeskkond, mis ühendab tehnilised lahendused ja kasutuskogemuse ühtseks tervikuks. Töö väärtus seisneb lisaks praktilisele

rakendusele ka rakendatud tehnoloogiate ja arendusmustrite analüüsis, mis võib pakkuda suuniseid sarnaste projektide loomiseks tulevikus.

Loodud mängu kood on saadaval aadressil <https://github.com/OPuusalu/gold-gold-gold/>.

Viidatud kirjandus

- [1] Ahn, L. von, & Dabbish, L. (2008). *Designing Games with a Purpose: Principles and Practices*. *Communications of the ACM*, 51(8), 58–67.
- [2] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [3] Golde, M., Argade, D., & Pawar, M. V. (2016). *Comparative Study and Analysis of Multiplatform Mobile Application Development*. *International Journal for Modern Trends in Science and Technology*, 2(12), 73–77.
- [4] Harley A. (2014). *Icon usability*. Nielsen Norman Group. <https://www.nngroup.com/articles/icon-usability/>
- [5] Karacali, H., Donum, N., & Cebel, E. (2024). *Cryptographic Enhancement of Named Pipes for Secure Process Communication*. *The European Journal of Research and Development*, 4(2), 1–18. <https://doi.org/10.56038/ejrnd.v4i2.428>
- [6] Kreinin, I. (2024). Kasutajaliidese disaini loomine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele (Bakalaureusetöö, Tartu Ülikool, Arvutiteaduse instituut).
- [7] Microsoft. *Create a cross-platform app with .NET MAUI*. Microsoft Learn, 2024. <https://learn.microsoft.com/en-us/training/modules/build-mobile-and-desktop-apps/> (vaadatud 07.12.2024).
- [8] OÜ Ziraff (2014). KASUTAJASÕBRALIKE E-TEENUSTE DISAINIMINE MAANTEEMETI NÄITEL.
- [9] World Wide Web Consortium (W3C). (2012). *Veebisisu juurdepääsetavuse juhised (WCAG) 2.0* (Eesti keeles). <https://www.w3.org/Translations/WCAG20-et/> (vaadatud 10.05.2025).

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Oliver Puusalu**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Platvormile AUR+ töölauarakenduse ja integratsioonisüsteemi loomine**, mille juhendaja on Jaan Janno, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Oliver Puusalu

15.05.2025