

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Aivaras Nakvosas

**Leidyklos paslaugų valdymo informacinė sistema**

**Publishing Service Management Information System**

Baigiamasis bakalauro darbas

Programų inžinerijos studijų programa, valstybinis kodas 6121BX023

Informacijos sistemų studijų kryptis

Vilnius, 2022

**Turinys**

[Iliustracijių sąrašas 3](#_Toc94044013)

[Lentelių sąrašas 4](#_Toc94044014)

[Santrumpų ir terminų žodynas 5](#_Toc94044015)

[Įvadas 6](#_Toc94044016)

[1. Leidyklos veiklos analizė ir informacinių sistemų apžvalga 7](#_Toc94044017)

[1.1. Leidyklos veiklos aprašymas 7](#_Toc94044018)

[1.2. Leidyklos leidinių gamybos proceso aprašymas 8](#_Toc94044019)

[1.3. Leidyklos informacinių sistemų apžvalga 11](#_Toc94044020)

[1.3.1. Apžvelgtų sistemų palyginimo lentelė 15](#_Toc94044021)

[1.4. Išvados 16](#_Toc94044022)

[2. Informacinės sistemos kūrimo technologijų analizė 16](#_Toc94044023)

[2.1. Serverinio dalies programavimo kalbų analizė 16](#_Toc94044024)

[2.2. Kliento dalies programavimo kalbų analizė 18](#_Toc94044025)

[2.3. Duomenų bazių analizė 19](#_Toc94044026)

[2.4. Išvados 19](#_Toc94044027)

[Literatūros sąrašas 20](#_Toc94044028)

# Iliustracijių sąrašas

[**1 pav. Leidinio gamybos procesai** 8](file:///C:\Users\Aivaras\Desktop\Aivaras%20Nakvosas%20Prifs%2018-6%20BD1.docx#_Toc94043511)

[**2 pav. Veiklos biudžeto pavyzdys [7]** 10](file:///C:\Users\Aivaras\Desktop\Aivaras%20Nakvosas%20Prifs%2018-6%20BD1.docx#_Toc94043512)

[**3 pav. Comma.cloud vaizdas [8]** 12](file:///C:\Users\Aivaras\Desktop\Aivaras%20Nakvosas%20Prifs%2018-6%20BD1.docx#_Toc94043513)

[**4 pav. Consonance vaizdas [9]** 13](file:///C:\Users\Aivaras\Desktop\Aivaras%20Nakvosas%20Prifs%2018-6%20BD1.docx#_Toc94043514)

[**5 pav. Schilling vaizdas [10]** 14](#_Toc94043515)

# Lentelių sąrašas

[**1 lentelė Sistemų vertinimo kriterijai** 11](#_Toc94043516)

[**2 lentelė Sistemų palyginimo lentelė** 15](#_Toc94043517)

# Santrumpų ir terminų žodynas

Serverinė dalis – tinklapio dalis, kuriame siunčiama turinio užklausą, serveris ją apdoroja ir sukuria atsakymą.

Kliento dalis – tinklapio dalis, kurią mato vartotojas ir į kurią tiesiogiai įsitraukia.

API – aplikacijų programavimo sąsaja, per kurią skirtingos sistemos gali apsikeisti duomenimis.

XML – žymėjimo kalba.

DB – duomenų bazė.

JS – „JavaScript“ programavimo kalba.

TS – „TypeScript“ programavimo kalba.

# Įvadas

Norint tapti autoriumi, labai svarbu įsigyti leidyklų paslaugas. Leidėjai yra tam, kad priimtų autoriaus rankraščius ir išleistų leidinius. Jie reguliuoja visą leidinių pardavimo žmonems ciklą: peržiūri pateiktus juodraščius, dirba su autoriais, kad juodraščiai būtų patobulinti, pateikia leidinių dizainą ir išspausdina fizinę kopiją. Nuo proceso pradžios iki pat pabaigos leidėjas investuoja daug laiko, pinigų ir patirties, siekdami užtikrinti, kad kiekvienas leidinys būtų pati geriausias, koks tik gali būti. Ir viskas daroma siekiant, kad autorių darbai patektų į kuo daugiau skaitytojų.

Leidinių užsakymo procesą sudaro: įsigijimas, redagavimas, dizainas, koregavimas, spausdinimas, pardavimai. Leidyba iš tikrųjų yra sudėtingas ir kelių etapų procesas, todėl nenuostabu, kad priklausant nuo leidinio tai gali užtrukti mažiausiai vienerius metus. Tačiau šis laikas galėtų būti sutrumpintas, naudojant sukurtą informacinę sistemą, kuri leistų vartotojams naudoti patogią apdorojimo sistemą nuo leidinių rankraščių pateikimo iki leidinių paskelbimo.

**Darbo objektas** – Leidyklos paslaugų valdymo informacinė sistema.

**Darbo tikslas** – Patobulinti leidyklos leidinių paruošimo spaudai sukuriant informacinę sistemą

**Darbo uždaviniai:**

1. Išanalizuoti leidyklos veiklą ir atlikti informacinių sistemų apžvalgą.
2. Palyginti informacinių sistemų kūrimo technologijas.
3. Pateikti kuriamos leidyklos paslaugų valdymo reikalavimų specifikaciją;
4. Suprojektuoti ir sukurti leidyklos informacinę sistemą.
5. Patikrinti ar sukurta sistema atlieka nurodytus reikalavimus.

**Darbo naujumas bei praktinė jo vertė**

Šiuo metu mažai egzistuoja valdymo sistemų, kurios orientuotųsi į leidyklų paslaugas. Egzistuojančios sistemos pagrinde dėmesį telkia į knygas. Todėl sukūrimas naujos sistemos, kuri pritaikytų įvairių leidinių rankraščių (knygų, žurnalų, laikraščių) pavertimą į galutinę versiją, jų darbo eigos stebėjimą ir valdymą supaprastintų ir palengvintų santykius tarp leidyklos ir autorių.

# Leidyklos veiklos analizė ir informacinių sistemų apžvalga

## Leidyklos veiklos aprašymas

Leidykla – tai įmonė, kuri perima autoriaus rankraštį (angl. manuscript), paverčia jį į leidinį ir pateikia jį plačiajai visuomenei. Leidiniai gali būti knygos, laikraščiai, žurnalai. Leidyklos labai skiriasi. Jos gali būti didžiulios arba mažos, pelningos, nuostolingos arba nesiekiantis pelno, jos gali būti visiškai skaitmeninės arba beveik vien spausdinamos, pasauliniai arba orientuoti į labai vietinę rinką [1]. Tačiau procesas tarp visų yra labai panašus.

Leidykla labai priklauso nuo rinkos sąlygų, yra sunkiai prognozuojama ir nepastovi. Leidyklos materialinės ir techninės bazės ypatumas yra tas, kad ji orientuota į specializuotą leidinių produktų pobūdį - knygos, vaizduojamoji menai, laikraščiai, žurnalai ir t.t [2]. Leidyklos turi arba samdo redaktorius, renka redakcines tarybas ir vertintojus, o tada suteikia jiems galimybę atlikti leidyklai gyvybiškai svarbų darbą, kuris deda užtikrinti pateiktų rankraščių pagrįstumą ir vientisumą. Leidyklos rengia ir spausdina priimtus darbus, tada platina juos prenumeratoriams arba pirkėjams, taip suteikdami galimybė dalytis informacija [3].

Leidyklos skyrius dažniausiai sudaro: redakcijos, sutarčių, kūrybos arba projektavimo, rinkodaros, finansų ir operacijų, gamybų skyriai. Svarbu paminėti, kad [1] teigia, jog leidyklos tiksliai neatspindės šių skyrių ir struktūrų, ir kad daugelyje mažų leidyklų visas šias funkcijas atlieka tik vienas ar du darbuotojai. Leidyklos redaktoriai atlieka visas pareigas, reikalingas leidinius įsigyti ir redaguoti, kad jos būtų išleistos. Jie taip pat bendrauja su leidinių agentais ir autoriais [4]. Kai tik užsakomas naujas leidinys ir pasirašoma sutartis, redaktoriai ir jų padėjėjai taip pat yra atsakingi už ryšio palaikymą su autoriumi rašymo proceso metu, kuris gali trukti kelias savaites, kelias, mėnesius ar kelerius metus [1]. Kadangi leidyba yra verslas, susijęs su intelektine nuosavybe, autoriaus sutartis yra svarbi ir lemiama dalis. Dėl šio teisinio leidybos proceso elemento sutarčių skyrius yra labai svarbus bendradarbiaujant su redaktoriais ir leidinių autoriais, kai reikia derėtis dėl sąlygų, pavyzdžiui, leidybos teisių, avansų, honorarų, terminų ir kitų teisinių klausimų [4]. Sutarčių skyriaus darbas iš pirmo žvilgsnio atrodo paprastas, tačiau dažnai gali būti sudėtinga. Tai svarbus darbas: leidyklos finansinė būklė ateityje priklauso nuo sutarčių, kuriomis užtikrinama jos intelektinė nuosavybės tvirtumas [1]. Dizaino departamente vadovas ir jo dizaineriai sukuria viršelį, kuris kartu su leidinio pavadinimu sudaro pirmąjį svarbų įspūdį vartotojui apie leidinį [1] [4]. Rinkodaros skyrius yra pagrindinė bet kurios leidyklos komanda. Bet kurios rinkodaros komandos pirmas darbas - nustatyti pagrindines leidinio auditorijas, o tada sukurti tinkamus pranešimus, kad leidinys būtų paruoštas pasiekti šias auditorijas, kurios gali gali būti pačios leidyklos prekybos atstovai, pardavėjai, bibliotekininkai ir pavieniai vartotojai [1]. Finansų skyrius turi stebėti, kaip išleidžiami ir gaunami pinigai.. Komanda tvarko gaunamas sąskaitas faktūras ir mokėjimus iš spaustuvių, knygynų, platintojų ir t.t., taip pat rūpinasi mokėjimais autoriams ir agentams. Operacijų skyrius rūpinasi atsargų valdymu. Bet kurią dieną privaloma stebėti klientų užsakymus, lyginant juos su prognozėmis, ir atlikti visus būtinus spausdinimo kiekių koregavimus tirti pagrindinių klientų dienos ar savaitės pardavimų duomenis, kad išsiaiškintų, ar atsargų lygis yra pakankamas patenkinti didesnių atsargų poreikį. Gamybos skyrius savo pastangomis ir glaudžiai bendradarbiaudamas su plačiu spaustuvių, įrišėjų, platintojų, dizainerių, tekstų rinkėjų, redaktorių, rodyklių sudarytojų ir kt. tinklu, gamybos komanda stengiasi, kad kiekvienas rankraštis taptų baigtiniu leidiniu kurioje nebūtų klaidų, o jo išvaizda ir įspūdis puikiai atitiktų turinį. ir pateisintų savo kainą [1]

## Leidyklos leidinių gamybos proceso aprašymas

Diagram

Description automatically generatedStandartinis leidinių gamybos procesas leidyklose prasideda nuo užsakymo gavimo fazės. Pabaigęs knygos, žurnalo ar kito pobūdžio leidinio rankraštį, autorius dažniausiai kreipiasi į agentą, kad padėtų „parduoti“ leidinį leidyklai. Agentas paprastai turi daugiau žinių apie komercinę leidinio vertę nei autorius ir yra įgudęs derėtis su leidykla . Leidybos įmonė gauna užsakymą iš kliento ir parengia pirminiam susitikimui, kad pagal kliento pageidavimus būtų nustatyta pagrindinė koncepcija. Jei leidykla domisi autoriaus darbu, jis gaus susitikimą su leidyklos komanda.

**1 pav. Leidinio gamybos procesai**

Kai autorius pristato rankraštį, redakcinė komanda turi jį perskaityti ir tada pasiūlyti autoriui bet kokius struktūrinius pakeitimus. Autoriai į tai atsako įvairiais būdais į tokius redagavimus – vieni džiaugiasi, kad leidėjai taip pakeičia savo darbus, kiti – mažiau [1]. Vadinasi, greičiausiai bus praleidžiama keletą savaičių dirbant su redaktoriumi, kad būtų užtikrinta, jog turinys geriausiai atitiktų leidyklos ir autoriaus leidinio viziją. Reikalaujamas redagavimo lygis priklausys tik nuo leidinio pobūdžio. Remiantis pagrindine koncepcija, parengtais pasiūlymais ir po serijinių derybų su klientu sudaromas susitarimas. [5]. Kai abi puses tenkina sąlygos, pasirašoma sutartis. Svarbu atkreipti dėmesį, kad leidyklos gali būti didelės ir įvairūs skyriai ir komandos egzistuoja leidykloje, kurios turi įvairių funkcijų ir paskirčių, kurie jau buvo paminėti anksčiau. Akivaizdu, kad gali atsirasti trikdžiai leidyklos procesuose. Pasak [6] leidybos procesas gali būti varginantis. Nuolat tenka daryti rankraščių pataisymus ir pirmyn ir atgal, taip pat pastoviai reaguoti į redagavimo užklausas. [6] taip pat teigia, kad prasta komunikacija tarp leidyklos ir autorių yra problema. Apklausus apie 800 autorių, 75 proc. jų nurodė, kad leidėjas niekada neprašė autoriaus pateikti atsiliepimų. 28 proc. autorių manė, kad bendravimas su leidėju buvo nenuoseklus leidinio išleidimo metu ir po jo. Tačiau [6] autorius ne iš karto prieina prie išvados, kad tai leidyklos kaltė, tačiau tai yra sunkumas neatsilikti, jei agentūros vis dar naudoja senamadiškus metodus ryšiams pavyzdžiui el. paštai su tokiu dideliu kiekiu kontaktų palaikyti.

Kai redaktorius ir autorius susitaria dėl galutinės rankraščio versijos, tai yra redakcinės komandos vaidmuo perduoti ją vadovaujančiai redakcijai, kad jie galėtų atlikti korektūrą, indeksuoti [1]. Indeksavimas yra labai svarbi negrožinio leidinio dalis; gerai sudarytas indeksavimas padeda skaitytojui rasti konkrečią informaciją apie tam tikras temas ir dalykus greitai ir paprastai būdu [7]. Korektūros procesas pastebės gramatines, temos klaidas, kurios gali būti nepastebėtos tų, kurie yra labai įsitraukę į redagavimo procesą. Kai kuriais atvejais autorius taip pat turėtų paprašomas, kad patikrintų galutinę versiją [7].

Dizaino etape projektavimo komanda parengia pagrindinį projektą, remdamasi pasiūlymais ir pagrindine koncepcija. Pagrindiniame dizaine yra teksto ir likusių meninių dalių, tokių kaip grafika, paveikslėliai ir visa papildoma reikalinga medžiaga, derinys [1]. Čia vyksta pokalbiai apie teksto dizainą, bet kokių vaizdų kokybę naudojamą leidinyje, todėl reikalingas, autoriaus galimybė atsakyti klausimų ir daug daugiau [1]. Dabar leidykla dirba siekdami užtikrinti, kad leidinys atrodytų paruoštas, kai jis bus išspausdintas. Kartais atmetami visi pradiniai dizainai arba vienas iš jų laikomas perspektyviu ir siunčiamas tolesniam darbui. Galiausiai, gavus ir autoriaus indėlį, nors tai gali būti nelengva, susitariama dėl galutinio dizaino, ir dizainerių darbas yra beveik baigiamas [1]. Autorius išsirinka viršelio dizainą, šriftą ir kažkas iš leidyklos surenks leidinį taip, kad tai atrodytų puikiai. Atlikus dizainą, prasideda leidinio gamybos procesas. Tačiau leidinio dizaino procesas irgi gali užtrukti. Dizaino detales, pavyzdžiui, šriftą ir paveikslėlius, dažnai reikia greitai pakoreguoti. Failų siuntimas pirmyn ir atgal, dokumentų formatavimas ir tolesnė komunikacija gali užimti leidėjų darbo valandas [6].

Svarbu laikytis terminų įvairiuose leidybos proceso etapuose, labai svarbu sudaryti tvarkaraštį .Leidybos projektuose gali dalyvauti daug žmonių ir kiekviename etape atlikti daug darbo. Todėl tvarkaraščiai turėtų būti realistiški. Neretai visas procesas, pradedant nuo pradiniu rašto iki galutinio leidinio, užtrunka vienerius metus. Laiko terminas, turėtų būti nustatytas šitiems dalykams: autoriaus pataisyta galutinė rankraščio versija, korektoriaus peržiūrėta ir pataisyta versija, atliktas dizainas ir maketavimas, spausdinimas, publikavimo data [7].

Graphical user interface, application, table

Description automatically generatedBiudžetas ir planuojamos veiklos (arba pajamų) ataskaita yra esminiai įrankiai rengiant leidinį. Juos reikėtų sudaryti prieš pradedant redakcinį procesą ir nuolat tikrinti ir prireikus atnaujinti per visą leidybos procesą [7]. Su autoriumi turi būti aptarta apmokėjimo ar kompensacijos forma ir nurodyti sutartyje [7]. Autorinis atlyginimas autoriui mokamas už kiekvieną parduotą leidinio egzempliorių. Autorinis atlyginimas paprastai autoriui mokami kartą per metus. Jie sudaro procentinę visuomenės kainos ir dėl jų turi būti deramasi tarp leidyklos ir autoriaus (paprastai 5-12%) [7]. Kalbant apie gamybos sąnaudas, Dėl maketavimo išlaidų paprastai mokama už puslapį. Spausdinimo išlaidos yra susijusios su tiražo kiekiu, kuris turėtų būti kuo tiksliau nustatytas, kiek egzempliorių turi būti išspausdinta ir išplatinta, kad būtų išvengta perteklinių atsargų. Atsargos turi būti stebimos, kad būtų galima nustatyti leidinio paklausą ir prireikus užtikrinti pakartotinį spausdinimą [7].

**2 pav. Veiklos biudžeto pavyzdys [7]**

## Leidyklos informacinių sistemų apžvalga

Šiame skyriuje bus apžvelgiama panašias egzistuojančias leidėjams skirtas leidyklos informacines sistemas, kurios apima visus valdymo aspektus nuo leidinio pradžios iki jo išleidimo. Taip pat pateikiama kriterijai į ką bus atsižvelgiama analizuojant sistemas.

**1 lentelė Sistemų vertinimo kriterijai**

|  |  |
| --- | --- |
| **Kriterijai** | **Aprašymas** |
| Leidinio projekto valdymas | Ar sistemoje galima kurti leidinių projektus, importuoti rankraščius, viršelius, priskirti prie projekto žmones ir matyti ką jie daro. |
| Užduočių planavimas ir terminai | Ar sistemoje galima kurti pagal leidyklos poreikius užduotis ir priskirti jiems atlikimo terminus. |
| Sutarčių ir autorinių atlyginimų valdymas | Ar sistemoje leidžiama kurti ir keisti sutartis tarp autoriaus ir leidyklų, valdyti nustatytas autorinių atlyginimų parametrus. |
| Komunikacija tarp vartotojų | Ar sistema turi bendravimo funkcionalumą, kurioje leidyklos darbuotojai ir autoriai galėtų diskutuoti. |
| Darbo eigos vizualizavimas | Ar sistema turi darbo procesų, užduočių atlikimo terminų, projekto progreso pavaizdavimą tam tikrais grafiniais elementais, kad perduoti informaciją vartotojui būtų aiškiau. |
| Leidinių specializacija | Kokius leidinius sistema leidžia kurti. |
| Kaina | Kokia kaina privaloma sumokėti, kad įsigyti sistemą. |

**Comma.cloud**

Leidėjams skirta projektų valdymo sistema „COMMA“ užtikrina, kad visos dalyvaujančios šalys būtų informuojamos apie pažangą ir apžvelgiami visi aktyvūs projektai. „COMMA“ siūlo autoriams būtiną paramą ir informaciją, įskaitant debiutuojančių autorių mokymą. Sistema siūlo integruotą pagalbą pardavėjams, joje yra automatinės funkcijos, skirtos pranešimams spaudai, sutartims, taip pat spaustuvių pasiūlymams surinkti.

Graphical user interface, application, email

Description automatically generatedSistema yra programinė įranga kaip paslauga (angl. Software-as-a-Service), kuri suteikia tikslią visų aktyvių knygų projektų apžvalgą, galima pamatyti koks yra jų progresas (*žr. 4 pav*.).

**3 pav. Comma.cloud vaizdas [8]**

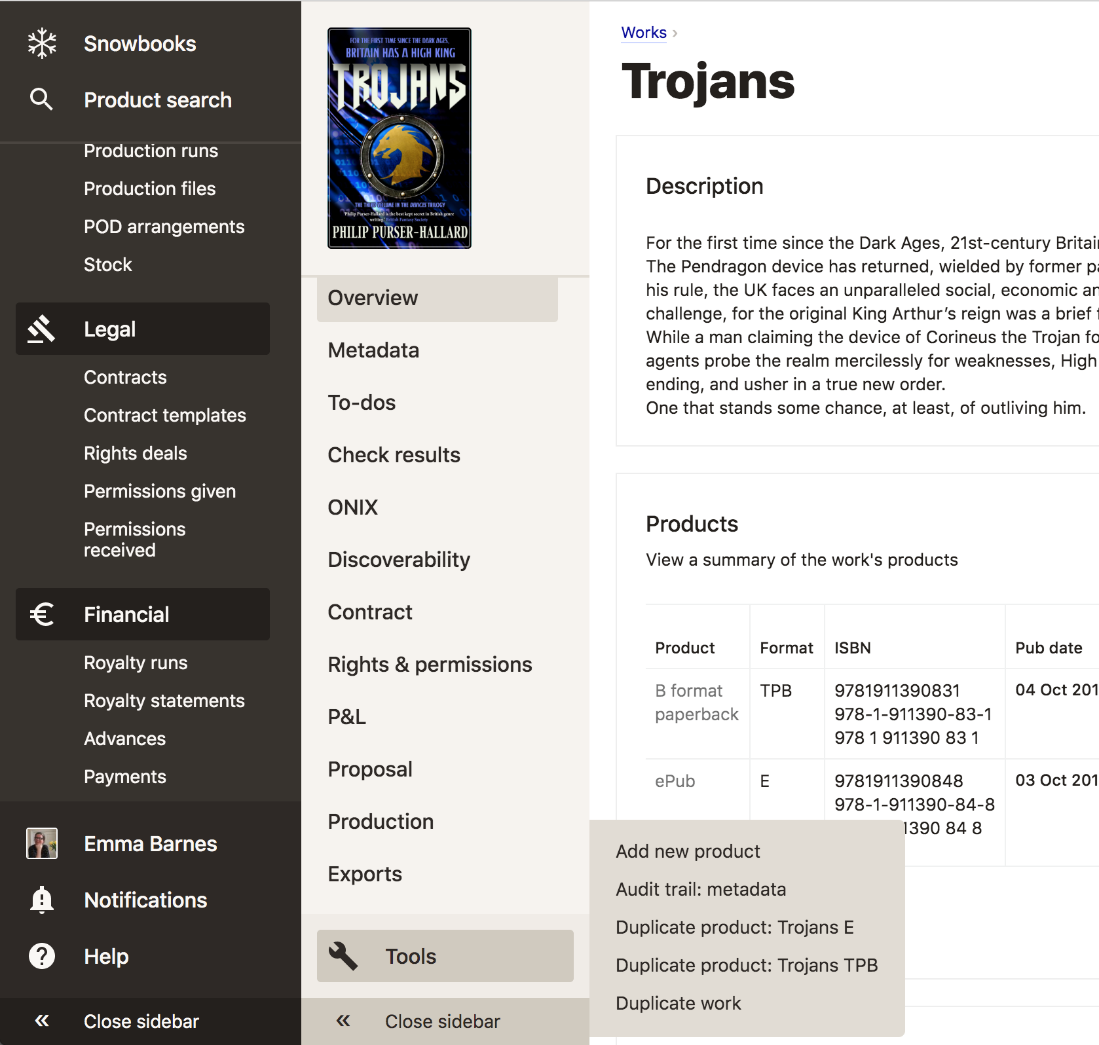
Darbuotojams skiriamos iš anksto nustatytos užduotys, atsižvelgiant į projekto žanro tipą ir jų vaidmenį leidykloje. Sistema padeda darbuotojams turėti aiškių užduočių, kurias reikia atlikti vaizdą, taip pat automatiškai siunčia priminimus, kai artėja užduočių atlikimo terminas.

Sistema leidžia importuoti įvairius failus pavyzdžiui tam tikros iteracijos knygos rankraštį susijusias su užduotim, tačiau nėra funkcijos kurti diskusijų, kurioje būtų galima aptarti atlikto darbo teigiamus ar neigiamus rezultatus. Sistema sudaro sutartis tarp autorių ir leidyklų, sutartis automatiškai tampa prieinami tiek darbuotojams, tiek autoriui, kad juos būtų galima lengvai pasiekti.

Kalbant apie finansus, sistema taip pat siunčia kainų pasiūlymus (angl. price quote) spaustuvėms. Šiuos pasiūlymus galima lengvai peržiūrėti, kad būtų galima palyginti. Pardavimų duomenys eksportuojami per „Excel", o autorinis atlyginimas autoriui siunčiamas el. paštu. Sistema kasdien rodo knygos projekto išlaidas ir pajamas.

Informacijos srauto pagrindas yra pranešimų duomenų bazė, kurioje automatiškai siunčiami tam tikri iš anksto nustatyti pranešimai visiems su projektu susijusiems asmenims, įskaitant autorių. Sistema sukurta naudojant prisitaikantį dizainą (angl. responsive design). Ji lengvai pasiekiama net mobiliuosiuose telefonuose ir planšetiniuose kompiuteriuose, vartotojo sąsaja yra patogi ir aiški.

**Consonance**

Kita leidybos valdymo sistema yra „Consonance“, kuri turi knygos įsigijimo, metaduomenų, sutarčių valdymo, autorinių atlyginimų, kontaktų valdymų modulius . Sistema taip pat irgi yra programinė įranga kaip paslauga.

**4 pav. Consonance vaizdas [9]**

Sistema leidžia nustatyti darbų atlikimo terminus, susijusius su publikavimo datomis ir sutarties duomenimis. Tai leidžia apibrėžti leidyklos darbo eigą pagal savo poreikius, bet kokiu jos verslui tinkamu detalumo lygiu. Tačiau pačioje sistemoje nėra nustatytų darbų atlikimo terminų vizualizacijos t.y grafinio išdėstymo ir darbuotojui gali pasidaryti sunku atsižvelgti į tam tikrų procesų užbaigimo terminus kai knygos projektas jų turi daug, bet integruojant trečios šalies kalendoriaus programelę (angl. third-party app) sistema automatiškai pateikia per ją atlikimo terminus vartotojui vizualiai matyti, taip pat sistema nerodo knygos pabaigtumo progresą.

Sistema suteikia matyti leidyklos knygų projektus ir kokias patikras tam tikra knyga nepraeina. Sistema nuolat siunčia pranešimus jeigu įvyksta pokyčiai knygos projekte. Sistema yra centralizuota, todėl leidžiama saugoti savo sutarties duomenis, avansus, rankraščių pristatymo datas, informacija apie teises sistemoje. Taip pat sistema automatiškai sugeneruoja sutartis. Sistemoje be knygos metaduomenų įvedimo, taip pat leidžiama importuoti įvairius failus.

Egzistuoja diskusijų kūrimas tarp projekto darbuotojų, suteikiant jai pavadinimą ir aprašymą. Diskusijoje galima pažymėti vartotojus ir aptarti visus kūrinio publikavimo aspektus. Tačiau šios diskusijos nėra priskiriamos prie tam tikro knygos gamybos etapo, visos sukurtos diskusijos egzistuoja prie projekto diskusijų skilties.

**Schilling**

„Schilling“ sistema teikia įvairius modulius, pavyzdžiui, turto, teisių, autorinių atlyginimų ir leidinio gyvavimo ciklo valdymo bei autorių portalo, palengvinančio bendravimą su autoriais ir jų atstovais, leidžiančius klientams supaprastinti darbo procesus. Sistema leidžia sukurti knygos projektą, kurioje galima importuoti failus pavyzdžiui rankraščius, viršelius (*žr. 6 pav.*). Sistemoje naudojami projektų šablonai, kurie užtikrina, kad būtų užpildyta daug informacijos iš anksto apie tam tikrus žanrus ir produktų tipus.

Graphical user interface, text, application, website

Description automatically generated

**5 pav. Schilling vaizdas [10]**

Sistemoje galima paskirstyti projekto vaidmenis ir suteikti atskirus darbuotojus, kurie yra aiškiai supažindinami su jų užduotimis, kad jie visada būtų informuoti apie savo atsakomybę ir terminus. Kiekvienas gali matyti, ką iš darbuotojo tikimasi daryti. Pavyzdžiui, kai redaktorius kvalifikuoja projektą, jis automatiškai siunčiamas vyriausiajam redaktoriui. patvirtinimui, o tada teisių skyrius gauna pranešimą, kad jie gali pradėti rengti sutartį, tačiau sistemoje irgi trūksta užduočių, atlikimų terminų ir progreso vizualizacijos.

Kai paskelbiama knyga, sistema surenka visą informaciją apie gaminį, kuri turėtų būti naudojama rinkodarai. Galima matyti pardavimo sąmatą, kuri užtikrina, kad bus galima turėti realias pajamų lūkesčius. Sistema palaiko visus įsigijimo proceso etapus pirmąsias derybas, sutarties pasirašymą. Galima kurti savo šablonus sutarčiai. Kai susitarimai jau yra sistemoje, autoriniai atlyginimai už kiekvieną pardavimą automatiškai atsiskaitoma su iš anksto nustatytais tarifais, kainomis.

Sistema turi komunikacijos portalą, kuriame galima įkelti sutartis ir suteikti autoriams galimybę jas peržiūrėti, pateikti pastabas, spausdinti ir net pasirašyti sutartis bei siūlyti pakeitimus internetu.

Didžiausias „Schilling“ minusas tai, kad sistema yra padalinta į modulius. Tai sudaro autorių teisių valdymas, autorinių atlyginimų valdymas, komunikacijos portalas, leidybos ciklo valdymas ir juos reikia įsigyti atskirai. Pavyzdžiui, kad įkelti sutartis ir suteikti autoriams galimybę jas peržiūrėti, pateikti pastabas, prireiks turėti komunikacijos portalą, todėl turint vieną modulį leidybos procesas įmonei gali nepalengvėti

### Apžvelgtų sistemų palyginimo lentelė

Šiame skyriuje bus pateikta sistemų funkcionalumų palyginimų lentelė pagal anksčiau pateiktus kriterijus (žr. *1 lentelė*).

**2 lentelė Sistemų palyginimo lentelė**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sistema**  **Kriterijai** | **Comma.cloud** | **Consonance** | **Schilling** |
| Leidinio projekto valdymas | + | + | + |
| Užduočių planavimas ir terminai | + | + | + |
| Sutarčių ir autorinių atlyginimų valdymas | + | + | + |
| Komunikacija tarp vartotojų | - | + | + |
| Darbo eigos vizualizavimas | + | - | - |
| Leidinių specializacija | Knygos | Knygos | Knygos |
| Kaina | Nenurodyta | $350.0/mėn | Nenurodyta |

## Išvados

Leidyklos gali būti didžiulės arba mažos, jos gali būti pasaulinio masto arba orientuotos į labai vietinę rinką, tačiau leidyklos įmonės struktūra iš esmės yra tokia pati. redakcijos, sutarčių, kūrybos arba projektavimo, rinkodaros, finansų ir operacijų, gamybų skyriai.

Tačiau leidyba iš tikrųjų yra sudėtingas ir kelių etapų procesas, tai sudaro rankraščio paruošimas ir jo tvarkymas su redaktoriais, tada leidinio dizaino projektavimas, klaidų tikrinimas ir galiausiai spausdinimas. Svarbu paminėti, kad gali pasitaikyti netikėtumų.bet kuriame etape. Autorius galbūt prieštaraus kopijų redagavimui, todėl dėl to gali būti veluojama, gali atsirasti tam tikros klaidos vėlesniuose etapuose, dėl ko visa tai gali būti sustabdyta ir procesas pradėtas iš naujo. Leidinių dokumentų siuntimas pirmyn ir atgal, jų formatavimas ir tolesnis bendravimas gali užimti leidėjų darbo valandas. Todėl kūrinio leidyba gali užtrukti ilgai.

Šiuolaikiniai leidėjai pereina prie skaitmeninių ryšių sistemų ir valdymo platformų, kad išvengtų problemų, susijusių su elektroniniais laiškais, peržiūros pastabomis ir kompiuterizuoti leidyklos darbo eigą. Kad nebereikėtų tvarkytis su krūvomis popierių ir vietoj to būtų centralizuota turinio saugykla, vietoj pirmyn ir atgal siunčiamų patikslinimų būtų įdiegta versijų kontrolė, kad būtų galima peržiūrėti ir redaguoti realiuoju laiku.

Yra jau egzistuojančių sistemų, kurioje leidyklos darbuotojams skiriamos iš anksto nustatytos užduotys, atsižvelgiant į leidinio projekto tipą ir jų vaidmenį leidykloje, kurti atlikimo terminus, susijusius su publikavimo datomis ir sutarties duomenimis, saugoti įvairius dokumentus, generuoti finansines ataskaitas. Tačiau kai kuriom sistemom trūksta papildomų funkcionalumų, pavyzdžiui darbo eigos vizualizavimo, kitokių leidinių gamyba arba egzistuoja sistemos už kurių funkcionalumą reikia papildomai mokėti įmonėms.

# Informacinės sistemos kūrimo technologijų analizė

## Serverinio dalies programavimo kalbų analizė

**Java**

Java yra daugiaplatformė statinė objektinė programavimo kalba. Ši kalba yra nemokama ir gali būti naudojama visose platformose, o tai reiškia, kad bet kokią vienoje platformoje parašytą programą galima lengvai perkelti į kitą platformą [11]. Tai lemia „Java" kompiliatorius, kuris nesukompiliuoja kodo konkrečiai mašinai. Vietoj to „Java" sukuria unikalų formatą, vadinamą baitkodu (angl. bytecode). Jis vykdomas pagal virtualiosios mašinos specifikacijoje nustatytas taisykles. „Java" sintaksė panaši į „C++“, tačiau „Java nesuteikia žemo lygio programavimo funkcijų, pavyzdžiui, rodyklių. Be to, Java kodai visada rašomi klasių ir objektų pavidalu [11]. „Java“ kodai paprastai yra lengviau prižiūrimi, nes „Java" neleidžia daugelio dalykų, kurie gali lemti blogą ir (arba) neefektyvų programavimą, jei naudojami neteisingai. Pavyzdžiui, kadangi nėra rodyklių, bloga prieiga prie atminties (angl. memory leak) negalima.. „Java“ taip pat turi šiukšlių surinktuvą (angl. garbage collector), kuris pastoviai tikrina ar yra nebenaudojami objektus ir juos pašalinti, kad būtų galima atlaisvinti vietos atmintyje. Nors tai yra objektinė kalba, „Java“ palaiko funkcinį programavimo stilių nuo „Java 8" versijos, o tai leidžia sumažinti programuotojui darbą ir parašyti funkcionalumą keliomis eilutėmis. „Java“ turi daug aplikacijų programavimo sąsajų (angl. Application Programming Interface, API). Galima naudoti daugiau kaip 4500 API, todėl programuotojai gali pasirinkti bet kurią pagal savo konkrečius reikalavimus. Šios API siūlo beveik viską - nuo XML apdorojimo iki DB jungčių [12].

„Spring“ yra turbūt populiariausias „Java“ karkasas (angl. framework), kuris skirtas teikti visapusišką infrastruktūros paramą „Java" programoms kurti. „Spring" pasirūpina infrastruktūra, kad programuotojas galėtų sutelkti dėmesį į savo programą [13]. Pagrindinis šio karkaso bruožai yra priklausomybės injekcija (angl. dependency injection, DI) ir valdymo inversija (angl. Inversion of control, IoC). Valdymo inversija - tai programinės įrangos inžinerijos principas, pagal kurį objektų ar programos dalių valdymas perduodamas karkasui [13]. Šiuo pagrindu valdymo inversija pasirūpina klasių priklausomybių injekcijomis, programuotojui tik reikia nurodyti klasių priklausomybes. Tačiau „Spring“ pasižymi labai didelių failų konfiguracijų kiekiu todėl egzistuoja „Spring Boot“, kuris yra „Spring“ karkaso pletinys. Jis pašalina minėta problema ir yra labai naudingas kuriant savarankiškas, didelio lygio taikomąsias programas su minimaliomis pastangomis. „Spring Boot“ yra itin patogus kurti svetaines, interneto paslaugas (angl. *web services*). Kitas itin taikomas karkasas yra „Hibernate“, kuris paverčia objektinį domeno modelį į reliacinę duomenų bazės lentelę. Šis karkasas yra lengvai integruojamas su „Spring Boot“.

**C#**

„C#“ yra į objektus orientuota ir saugos tipo programavimo kalba. "C#" sintaksės požiūriu yra labai panaši į „Java“. Abi kalbos yra orientuotos į objektus, todėl palaiko tokius principus kaip inkapsuliacija, polimorfizmas, paveldėjimas.. „C#“ taip pat turi šiukšlių surinktuvą. C# programos vykdomos naudojant .NET, virtualią vykdymo sistemą, vadinamą bendrąja kalbos vykdymo sistema (angl. common language runtime, CLR) ir klasių bibliotekų rinkinį. CLR - tai „Microsoft“ įgyvendinta bendrosios kalbos infrastruktūra (angl. common language infrastructure, CLI). CLI yra pagrindas kurti vykdymo ir kūrimo aplinkas, kuriose „C#“ ir bibliotekos sklandžiai bendradarbiauja [14]. „C#“ gali naudoti „ASP.NET“, kuris yra svetainių sistemų kūrimo karkasas. Kalbant apie skirtumus tarp „Java“ ir „C#“, „Java“ puikiai tinkama sudėtingoms svetainių sistemoms, pasižyminčioms dideliu vienalaikiškumu, kurti, o „C#“ idealiai tinka žaidimams ir mobiliesiems įrenginiams kurti. Kitas labai svarbus dalykas „Java“ bibliotekų ekosistema yra plati ir gerai dokumentuota, o tai dar labiau padeda sparčiai kurti programas, tuo tarpu „C#“ bibliotekos veikia „Microsoft“ ekosistemoje, kuri, palyginti su „Java“, yra ribota. Taip pat su „Java“ naudojami „Maven“ arba „Gradle“ įrankai, kurie leidžia lengvai tvarkyti bibliotekų priklausomybes ir kurti kompiliacijas, o „.NET“ turi gana ribotą ekosistemą.

## Kliento dalies programavimo kalbų analizė

**JavaScript**

„JavaScript“ (JS) yra prototipais pagrįsta, daugiaparadigminė, vienos gijos, dinaminė kalba, palaikanti objektinį, imperatyvinį ir deklaratyvinį (pvz., funkcinio programavimo) stilius [15]. JS yra interpretuojama, o ne kompiliuojama kalba, tai reiškia, kad naršyklė eilutė po eilutės interpretuoja JS kodą ir jį paleidžia. Priešingai, kad būtų galima paleisti kompiliuotą kalbos kodą, ją reikia sukompiliuoti Pagrindinė sintaksė sąmoningai panaši į „Java“ ir „C++“ [15]. Ši kalba labiausiai skirta dirbti su internetiniais puslapiais. JS veikia svetainės kliento pusėje ir gali būti naudojamas projektuoti ir (arba) programuoti, kaip svetainių puslapiai elgsis įvykus įvykiui [15]. JS yra dinaminė kalba. Tai reiškia, kad JS interpretatorius nereikalauja aiškiai deklaruoti kintamųjų tipus prieš juos naudojant. JS yra prototipais pagrįsta kalba, o tai reiškia, kad joje naudojamas prototipinis paveldėjimo modelis, o ne plačiai žinomas klasių paveldėjimas [15]. Tai reiškia, kad, užuot kūrę klases ir iš jų išvedę objektus, apibrėžiame objekto prototipą ir naudojame šį objekto prototipą, kad sukurtume daugiau to paties tipo objektų. JS turi didelės bendruomenės paramos. Tai užtikrina, kad yra daug atvirojo kodo bibliotekų, kuriomis galima nemokamai naudotis.

**TypeScript**

„TypeScript“ (TS) yra programavimo kalba, kurią sukūrė ir prižiūri „Microsoft“. Tai griežtas sintaksinis „JavaScript“ rinkinys, kuris į kalbą įtraukia neprivalomą statinį tipavimą [16]. TS palaiko tokias objektinio programavimo principus kaip klasės, interfeisus (angl. interface), paveldėjimas ir kt. Kadangi TS yra JS rinkinys, vadinasi parašytas TS kodas gali būti paleistas naudojant JS. TS visada nurodo kompiliavimo klaidas kodo kūrimo metu, nes TS kodą reikia kompiliuoti, o JS kodo nereikia.. Dėl to paleidimo metu klaidų tikimybė yra labai maža negu naudojant JS. Todėl TS yra patogesnis kuriant dideles taikomąsias programas, nes analizuoti tokių projektų kodą jeigu jie naudotų dinaminius tipus būtų labai sunku programuotojams.

## Duomenų bazių analizė

**MySQL**

„MySQL“ yra „Oracle“ sukurta reliacinė duomenų bazių valdymo sistema (angl. Relational Database Management System, RDBMS), pagrįsta struktūrizuotų užklausų kalba (angl. Structured Query Language, SQL). Kaip ir kitose reliacinėse duomenų bazėse, „MySQL“ duomenys saugomi lentelėse, sudarytose iš eilučių ir stulpelių. Nors „MySQL " dažnai siejama su interneto programomis ar svetainių paslaugomis, ji buvo sukurta taip, kad būtų puikiai suderinama su kitomis technologijomis ir architektūromis. duomenų bazė veikia visose pagrindinėse kompiuterių platformose, įskaitant „Unix" operacines sistemas, pavyzdžiui, daugybę „Linux“ distribucijų ar „Mac OS“, ir „Windows“ (mysql). „MySQL“ kliento ir serverio architektūra leidžia palaikyti įvairias serverinius programavimo kalbas ir skirtingas programavimo sąsajas. [17].

**Cassandra**

„Cassandra“ yra „Apache“ sukurta atvirojo kodo duomenų bazė. Tai „NoSQL“, kuris yra duomenų bazių projektavimo metodas, leidžiantis saugoti duomenis ir teikti užklausas ne pagal tradicines reliacinių duomenų bazių struktūras [18]. „Cassandra“ taip pat lanksčiai keičia duombazės dydį. Kadangi „Cassandra“ yra tokia dinamiška, duomenų bazę galima didinti arba mažinti pagal poreikį, kitaip nei „MySQL“, kuriai reikia prastovos. Tačiau „Cassandra“ trūkumas yra tas, kad dėl to, jog ji yra „NoSQL“, nėra struktūrizuotos „SQL“ sintaksės, todėl būtų galima sudaryti sąrašą funkcijų, kurių „Cassandra“ neturi. Pavyzdžiui, nėra duomenų vientisumo. Ypač tai, kad „Cassandra“ yra „NoSQL“, yra tikimybė susidurti su pasikartojančiais duomenimis, nes jie nebus automatiškai atmetami, kaip tai darytų „MySQL“.

## Išvados

Kuriamoje leidyklos procesų valdymo sistemoje bus naudojama „Java“ programavimo kalba, nes ji yra objektiškai orientuota su daugybė atviro kodo bibliotekų. „Java“ kalba turi tokius karkasus kaip „Spring Boot“ ir „Hibernate“, kurie žymiai palengvina kurti internetinė svetainė ir leidžia lengvai modeliuoti objektinius modelius į realicinę duomenų bazę. Taip pat svarbu paminėti, kad „Java“ turi didelį dokumentacijos kiekį. „TypeScript“ bus antroji naudojama programavimo kalba dėl savo statinių tipavimo ir objektinio programavimo savybių pritaikymo „JavaScript“ kalbai. Duomenų bazė bus „MySQL“ dėl struktūrizuotų užklausos kalbos duomenų manipuliacijos savybių ir duomenų vientisumo užtikrinimo.

# Literatūros sąrašas

1. Barker, D. 2020. The structure and workings of a publishing house. In Baverstock, A., Bradford, R. and Gonzalez, G. (eds.). Contemporary Publishing and the Culture of Books. Abingdon: Routledge, pp. 22-45. Prieiga per internetą: <https://www.taylorfrancis.com/chapters/edit/10.4324/9781315778389-2/structure-workings-publishing-house-david-barker>
2. Korinko, M., Zhelikhovskyi, G., 2020. The Problems and Challenges of Modern Publishing and Printing Industry of Ukraine. Informacijos mokslai 89, 8–16.. doi:10.15388/im.2020.89.36
3. Nahlen D, Clark S. The Publisher's Perspective on Journal and Book Publishing. *Semin Oncol Nurs*. 2018;34(4):381-385. doi:10.1016/j.soncn.2018.09.006
4. Peterson, V. The Top Departments of a Publishing House [žiūrėta 2022-01-25]. Prieiga per internetą: <https://www.thebalancecareers.com/major-departments-in-a-publishing-house-2800046>
5. Glykas, M., 2004. Workflow and process management in printing and publishing firms. International Journal of Information Management 24, 523–538.. doi:10.1016/j.ijinfomgt.2004.08.003
6. Swan, S. The Problems With Publishing: Stories From The Experts [žiūrėta 2022-01-08]. Prieiga per internetą: <https://blog.submittable.com/problems-with-publishing/>
7. ICOM. 2017. Publishing Manual. Prieiga per internetą: <https://icom.museum/wp-content/uploads/2018/07/Publishing-Manual_2017-EN.pdf>
8. „Comma.cloud“ [žiūrėta 2021-12-27]. Prieiga per internetą: <https://www.comma.cloud/en/presentation/>
9. „Consonance.app“ [žiūrėta 2021-12-28]. Prieiga per internetą: <https://www.consonance.app/>
10. „Schillingpublishing.com“ [žiūrėta 2021-12-28]. Prieiga per internetą: <https://schillingpublishing.com/>
11. GeeksForGeeks. Java Programming Language [žiūrėta 2022-01-23]. Prieiga per internetą: <https://www.geeksforgeeks.org/java/>
12. Bhatt, J. Why Choose Java: 7 Compelling Reasons to Learn the Programming Language [žiūrėta 2022-01-23]. Prieiga per internetą: <https://www.finoit.com/blog/7-reasons-learn-java-programming-language/>
13. Spring. Spring Framework Documentation [žiūrėta 2022-01-23]. Prieiga per internetą: <https://docs.spring.io/spring-framework/docs/current/reference/html/>
14. Microsoft. C# documentation [žiūrėta 2022-01-23]. Prieiga per internetą: <https://docs.microsoft.com/en-us/dotnet/csharp/>
15. Mozilla. JavaScript [žiūrėta 2022-01-23]. Prieiga per internetą: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
16. TypeScript. TypeScript [žiūrėta 2022-01-23]. Prieiga per internetą: <https://www.typescriptlang.org/>
17. MySQL. MySQL [žiūrėta 2022-01-23]. Prieiga per internetą: <https://www.mysql.com/>
18. Apache Cassandra. Open Source NoSQL Database [žiūrėta 2022-01-23]. Prieiga per internetą: <https://cassandra.apache.org/_/index.html>