

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Kursinis darbas

**Internetinė bendravimo ir įrankių valdymo sistema
"Talkie"**

Atliko:

Edvinas Strazdonis parašas

Rokas Buoželis parašas

Vadovas:

Dr. Joana Katina

Vilnius
2021

Turinys

Anotacija.....	3
Summary (Anotacija anglų kalba).....	4
Įvadas.....	5
1. Panašių sistemų analizė.....	7
1.1. Panašūs sprendimai užsienio rinkoje.....	7
2. Sistemos teorinė dalis.....	12
2.1. Sistemos funkcionalumas	12
2.2. Aktoriai ir objektai.....	13
2.3. UML diagramos	14
2.3.2. Klasų diagrama.....	16
2.3.3. Duomenų bazės modelis.....	18
3. Sistemos praktinis įgyvendinimas.....	20
3.1. Naudojamos technologijos.....	20
3.2. Sistemos funkcionalumas	20
3.3. Problemos	21
3.4. Programinis kodas.....	21
3.5. Saugumas	23
Išvados ir rekomendacijos	24
Literatūros sąrašas	26

Anotacija

Darbo tikslas – sukurti internetinę svetainę, kurios pagalba naudotojai galėtų lengvai pasiekti svetainės teikiamus įrankius bei laisvai ir nemokamai bendrauti su kitais svetainės vartotojais vaizdo ir garso perdavimo būdais. Taip pat yra siekiama, kad svetainė būtų naudojama kaip kasdienis tinklapis pamatyti viską kas tave domina vienoje vietoje, reiškiasi, kad vartotojas prisijungęs į sistemą norėtų naudotis ne vien tik nemokamais vaizdo pokalbiais, bet ir kitais įrankiais, kurie palengvintų ir sutrumpintų jo kasdienį naršymą internete, sumažintų skirtingų įrankių naudojimą. Svetainės įrankių kūrimas buvo vykdomas atsižvelgiant į vėliausias tendencijas bei išanalizuojant kitas panašaus pobūdžio sistemas. Darbo rezultatas yra sistema, kurios dėka vartotojas gali lengvai apturėti pokalbius su kitais vartotojais, taip pat vartotojas gali naudotis svetainėje esančiais įrankiais (Kripto valiutų kainų stebėjimas, Orai, Užrašinė, Naujienos, Kalendorius, „To do“ užrašinė) tokių būdu jam nereikia atskirai aplankyti kitų svetainių ar informacinių sistemų.

Summary

The main objective of this project is to create an internet application which allows users to communicate free of charge and use all tools provided by our system. Also, the other goal of this system is to be a website which people use in a daily bases not to only talk with people but also use tools so they wouldn't need to go the other websites and, in this way, user would shorten his time spent on the computer, because he wouldn't need to browse all over the internet to find tools that he needs on a daily basis. Tools provided by our website was carefully selected and we searched all over the internet to find what is trending on today's internet. Also, we made an analysis of similar websites and took a deeper look of what those applications could offer to their users besides communication. The result is a system that allows users to easily connect with other people across the world and have conversations with each other. Furthermore, our application provides tools that people can use in their daily basis such as the Weather section where people can check the current weather, Crypto tracker where people can view prices of their selected crypto currencies, Notebook where people can write their ideas and notes, News where people can subscribe to specific newspaper and they would get daily news from it, Calendar where people can mark their meetings and other important dates and last but not least we have a To Do notebook where people can write their jobs or other tasks. In the future we view our application as a bigger tool where people can decide which tools they need on a daily basis and they can add them on their pages, but this requires further analysis of other systems. The conclusion of this project is that we are trying to make fully done application where people could reach all their needs in one page.

Ivadas

Pabandykite įsivaizduoti šiuos laikus be informacinės sistemos įmonėje, kuri suteikia vienokį ar kitokį vaizdo bei garso perdavimo būdą. Darbas be šių sistemų būtų neįmanomas ypač atsižvelgiant į šių dienų aktualijas, kaip Covid-19 virusas, bei vis dažniau darbuotojų mėgstamas nuotolinis darbas. Taip yra būtent todėl, kad šios sistemos smarkiai palengvina įmonės veiklą, naudodamiesi šiomis sistemomis darbuotojai dirba greičiau ir efektyviau, bendravimas tarp žmonių ar skyrių tampa paprastesnis, nei gyvi susitikimai, nekalbant apie bendravimą tarp įmonių skyrių, kuriuos riboja valstybinės sienos. Tačiau šios sistemos dažnai būna labai vienus, kitaip tariant jos būna sufokusuotas tik į vaizdo ar į garso perdavimą, kartais pasidalinimą ekranu, tačiau dažnai kitų funkcijų jos neturi. Dėl šios priežasties kompanijos ir jos darbuotojai turi naudotis ir pirkti labai daug produktų ir sistemų, kuriomis taip pat reikia išmokti naudotis kas reiškia didesnes laiko sąnaudas bei didesnes išlaidas įmonėj. Daugelis įmonių šią problemą puikiai supranta, tačiau kurti savo sistemą, kurioje yra viskas vienoje vietoje taip kaip tau reikia kainuos galbūt ir tiek pat kiek atskiros programos, tačiau tai atims laiką ir vėl gi papildomus pinigus iš ko iškyla problema, kad įmonė savo sistemos, kur viskas yra vienoje vietoje taip ir neturi, o ir toliau naudoja kitų įmonių siūlomus atskirus produktus. Šiame darbe yra siekiama sukurti internetinę sistemą, kurioje vartotojas galėtų turėti pokalbius su kitais vartotojais ir taip pat turėtų prieigą prie dažniausiai naudojamų įrankių vienoje vietoje. Šiai sistemai įgyvendinti buvo nuspręsta kurti ją kaip žiniatinklio sistemą, nes jos pagrindinis funkcionalumas reikalauja interneto prieigos, taip pat šio pobūdžio sistemos vis dažniau ir sparčiau veržiasi į priekį. Taip yra todėl, kad internetinę sistemą gali pasiekti bet kas ir ja naudotis būna labai lengvai, taip pat tai sutaupo daug laiko, nes jos atskirai nereikia įdiegti į visų vartotojų kompiuterius, kas sumažina laiko bei galimų išskylančių sunkumų sąnaudas. Taip pat internetinė sistema yra lankstesnė suderinamumo klausimu, nes nereikia atsižvelgti į operacinę sistemą, vartotojo turimo prietaiso galingumą bei kitas klaidas, kurios gali iškilti bandant sistemą suderinti su įvairiomis operacinėmis sistemomis.

Kuriant šią sistemą mes turėjome daugybę skirtingų iššūkių, kurie mums padėjo pagilinti savo jau turimas ar naujai įgytas žinias internetinės aplikacijos kūrimo. Išmokome naudoti įvairias internetines technologijas, kaip Mongo duomenų bazių valdymo sistemą, Express.JS karkasą, įvairias Express.JS bibliotekas, Bootstrap karkasą, PeerJS ir Socket.IO bibliotekas, bei Node.JS karkasą skirtą kurti serverius su JavaScript kalba. Turėjome galimybę stipriai pagilinti žinias darbe grupėje.

Darbo tikslas – sukurti internetinę svetainę, kurios pagalba vartotojai galėtų lengvai pasiekti svetainės teikiamus įrankius bei laisvai ir nemokamai bendrauti su kitais svetainės vartotojais vaizdo ir garso perdavimo būdais. Taip pat yra siekiama, kad svetainė būtų naudojama kaip kasdienis tinklapis pamatyti viską kas tave domina vienoje vietoje, reiškiasi, kad vartotojas prisijungęs į sistemą norėtų naudotis ne vien tik nemokamais vaizdo pokalbiais, bet ir kitais įrankiais, kurie palengvintų ir sutrumpintų jo kasdienį naršymą internete, sumažintų skirtingų įrankių naudojimą. Šios internetinės

svetainės galutinis tikslas būtų turėti sistemą su gausiu įvairių įrankių pasirinkimu, kuriuos įmonės ar paprasti vartotojai galėtų išsirinkti ir susidėlioti patys, tai reikštų, kad vartotojas vienoje vietoje galėtų susikurti savo reikiama ir norimą aplinką, kas pašalintų naudojimąsi daug įvairių skirtingų programų, vartotojas sutaupytu daug laiko ir viską galėtų rasti vienoje vietoje. Šioje sistemoje vartotojai galės prisijungti, užsiregistruoti, kalbėti su kitais vartotojais, dirbti su įrankiais (Kripto valiutų kainų stebėjimas, Orai, Užrašinė, Naujienos, Kalendorius, „To do“ užrašinė).

Mūsų darbo uždaviniai yra išanalizuoti panašias sistemas, surasti šių sistemų trūkumus bei privalomus, iš kitų sistemų pasisemti idėjų bei dalykų kurių galbūt reiktų vengti. Taip pat susidaryti projekto planą į kurį įeitų: pagrindinės sistemos funkcijos, reikiamas funkcionalumas, išsiaiškinti kokios technologijos geriausiai tinka kurti šiam projektui (atsižvelgiant į turimas žinias), sukurti duomenų bazės modelį bei nubraižyti UML diagramas.

Pasiekti rezultatai paskutiniame mūsų sistemos variante: internetinė svetainė, kurioje vartotojas gali užsiregistruoti bei prisijungti, vartotojas laisvai ir nemokamai gali bendrauti su kitais sistemos vartotojais, vartotojas gali naudotis sistemos įrankiais. Įgyvendinti sistemos įrankiai: Užrašinė, kurioje vartotojas gali pasižymėti svarbius dalykus, kalendorius kuriame vartotojas gali žymėti susitikimus ir kitus dalykus, „To do“ užrašinė, kurioje vartotojas gali stebėti savo darbus bei juos pildyti, Orai, kur vartotojas gali stebėti dabartinio oro sąlygas, Kripto valiutų sekimo puslapis, kurioje vartotojas gali stebėti mėgstamas kripto valiutas bei naujienų polapis, kuriame vartotojas gali išsirinkti dienraštį iš kurio galės skaityti naujienas.

1. Panašių sistemų analizė

Panašių sistemų nepavyko rasti daug, tačiau po ilgo naršymo internete galiausiai atradome būdą rasti kelias sistemas, kurios bent keliais aspektais turėtų panašų funkcionalumą kaip ir mūsų sistema. Idealiu scenarijumi buvo ieškoma internetinių sistemų kurios turėtų tokį funkcionalumą: galimybė prisijungti bei užsiregistruoti, galimybė bendrauti su kitais sistemos vartotojais vaizdo bei garso perdavimo būdu, svetainė turėtų turėti bent mažą panašų įrankių pasirinkimą į mūsų, pavyzdžiui, užrašinė ar kalendorius. Tačiau tokių „idealaus scenarijaus“ svetainių nėra daug, todėl sistemų ieškojimui taip pat taikėme kriterijus, tačiau jie buvo labiau bendriniai: ieškomos sistemos turėjo turėti bendravimą vaizdo ir garso perdavimo būdu, taip pat svetainės turėjo turėti dar bent vieną iš anksčiau idealiame scenarijuje nurodytų funkcionalumų. Užsienio rinka tikrai turėjo ką pasiūlyti iš visiems gerai žinomų pavydžių, tačiau su Lietuviškomis sistemomis iškilo keblumų, todėl savo kriterijus turėjome smarkiai sumažinti, deja net sumažinus kriterijus iki minimumo panašių spendimų neradome. Galima svarstyti kodėl Lietuvoje tokio tipo svetainės nėra populiarios ir vienas iš atsakymų yra turint tokią mažą rinką, kuri ir taip naudojasi užsienio pateikiamais tokių sistemų variantais labai sunku sukurti kažką naujo ar pranašesnio, kad galėtum nukonkuruoti dideles kompanijas bei persivilioti klientus pas save.

1.1. Panašūs sprendimai užsienio rinkoje

Pirmoji analizuojama sistema yra „Google Meet“ – tai sistema skirta nemokamai bendrauti vaizdo ir garso perdavimo būdu. Šia sistema gali naudotis nemokamai visi, kurie turi savo Google paskyrą. Sistema suteikia vartotojui galimybę sukurti susitikimą vėliau, sukurti susitikimą iškart šiuo metu, ir leidžia susiplanuoti susitikimą naudojant „Google Calendar“ įrankį. Sistema savyje papildomų įrankių neturi, tačiau suteikia lengvą prieigą prie visų Google suteikiamų įrankių kaip, pavyzdžiui, „Docs“. Sistemoje yra realizuota patogi vaizdo ir garso perdavimo sistema, jeigu vartotojas nori, gali dalintis tik savo vaizdu arba garsu atskirai. Sistema yra labai draugiška vartotojui, viską ką vartotojas turi padaryti tai: kad prisijungtų į kito žmogaus sukurtą pokalbį įrašyti gautą kodą į laukelį ir paspausti JOIN, kad sukurtum pokalbį pats, tereikia paspausti New meeting ir pokalbis prasidės, pokalbiui prasidėjus gausi kodą, kurį galėsi nusiųsti žmonėms su kuriais nori kalbėti.

Antroji analizuojama sistema yra Microsoft Teams programa. Šiomis dienomis viena iš dažniausiai naudojamų sistemų bendravimui internete. Ši sistema yra pritaikyta įvairioms platformoms, tai reiškia, kad ją gali pasiekti visi norintys. Sistema yra skirta bendravimui tarp žmonių grupių ir daugiau, kitaip tariant, sistema suteikia galimybę bendrauti su vienu žmogumi atskirai, taip pat gali sukurti komandą ir kalbėti su daug žmonių vienu metu. Teams sistema suteikia galimybę dalintis failais ir juos saugoti savyje. Sistema taip pat turi puikią kitų programų integraciją, kuri leidžia lengvai pasiekti norimus

įrankius per Teams, tai puikus būdas netik išvengti nepasitenkinimo, kad sistema neturiu kažkokio įrankio, bet tai taip pat puiki kitų įmonių integracija į sistemą. Taip pat Teams platforma kaip įrankius siūlo naudotis Microsoft produktais, kaip, pavyzdžiui, Microsoft „Word“. Sistema nėra visiškai draugiška vartotojui, tai reiškia, kad naujas vartotojas turėtų prie sistemos priprasti ir išmokyti ją naudotis, kadangi pati sistema yra tikrai didelė ir siūlo daug skirtingų paslaugų.

Trečioji internetinė sistema kurią analizavome yra „Slack“, kitaip dar vadinama Microsoft „Teams“ konkurentinė aplikacija, kuri išliejo pagrindus dabartiniam Teams. Slack yra labai populiarinė internetinė vaizdo ir garso bendravimo sistema, kuri daugiausiai yra naudojama įvairios įmonės bendravimui tarp komandų. Slack dalinasi labai panašiais sisteminiiais sprendimais kaip ir Teams, kadangi jos yra konkurentės turėjome apžvelgti jas abi, kad galėtume nuspręsti, koks šių sistemų skirtumas ir ką mes galime pasiimti iš jų. Slack pagrindas išlietas ant bendravimo komandose ir turėjimo savo kanalo, kuriame gali būti įvairūs poskyriai bendravimui. Slack taip pat nėra visiškai draugiška naujiems vartotojams ir šia sistema reikia išmokyti naudotis, tačiau, jeigu prieš tai esate naudoję Teams tai nebus sunku, nes jos abi naudoja lygiai tokį pat išdėstymą ir funkcionalumą. Sistema turi tik kelis įrankius padedančius vartotojui pamiršti kitas sistemas tačiau tai nėra esminiai įrankiai, kuriais galėtum naudotis kasdien. Slack internetinė sistema siūlo „Workflow builder“, tai yra įrankis sutelktas į komandinį darbą ir dažniausiai naudojamas įmonėse, šis įrankis padeda siųsti įvairias užduotis bei prašymus kitiems komandos nariams. Kitas įrankis, kuris taip pat pasitarnauja labiau patiem vadovams nei paprastiems vartotojams yra Analytics, kur gali stebėti kanalo ir žmonių aktyvumą bei atliktas užduotis bei planus, kitus statistinius duomenis. Slack, mūsų nuomone, nusileidžia Microsoft Teams sistemai funkcionalumu ir įvairių įrankių skaičiumi, tačiau dėl mažesnio įvairių funkcijų kiekio šia sistema naudotis ir išmokyti valdyti yra lengviau.

Ketvirtoji internetinė sistema, kurią analizuojame yra „Discord“. Ši sistema yra gerai žinoma tarp jaunimo ir bendruomenių, kurios dažnai būna susifokusavusios į žaidimus. Sistemoje galima turėti kontaktus, kitaip vadinamus draugus, gali priklausyti serveriams arba bendruomenėms. Discord sistema suteikia vartotojams galimybę kurti savo pačių serverius į kuriuos gali kviesti žmones, kurti bendruomenės, bendrauti ir dalintis failais atskirai savo serveryje. Discord palaiko failų saugojimą todėl išsiųstas failas serveryje ar pokalbyje išliks ilgam. Discord suteikia galimybę dalintis ekranu bei kalbėti tiek vaizdo skambučiais, tiek balso skambučiais. Sistema suteikia galimybę vartotojams ir serverių administracijai redaguoti savo kanalus taip, kaip jie nori, reiškiasi, kad serverio viduje žmonės gali importuoti savo programinį kodą ir sukurti įvairius papildomus dalykus, ko pats Discord iš savęs nesiūlo. Sistema taip pat yra palaikoma visose platformose, reiškiasi, kad yra lengvai prieinama visiems vartotojams. Papildomų įrankių pati sistema nesuteikia, kitaip tariant, įrankius vartotojas gali pasiekti tik būdamas serveryje į kurį tam tikras įrankis bus įdiegtas, kitu būdu Discord nesuteikia galimybės naudotis kažkokiais papildomais įrankiais nei pokalbiai ir dalyvavimas bendruomenėse. Vartotojo sąsaja yra pakankamai lengvai suprantama, tačiau prireiks kelių bandymų

ir trumpo pasivaikščiojimo po aplikacija, kad suprastum kaip ji veikia. Sistema iš savęs yra intuityvi, nėra sunku suprasti ką reikia spausti, kad patektum į tam tikrą programos dalį.

Apibendrintą šios analizės dalį galima rasti pirmoje lentelėje, joje yra pateikiami išanalizuotų sistemų funkciniai sprendimai lyginant su mūsų sistema. Buvo analizuojami pagrindiniai dalykai, kuriuos turi mūsų sistema ir turi kitos sistemos. Antroje lentelėje galima rasti kiekvienos sistemos analizę apžvelgiant jos privalumus ir trūkumus.

Sistema/Funkcija	Vaizdo Skambučiai	Balso Skambučiai	Kalendorius	Duomenų saugykla	Darbų planavimo įrankis
Discord	Taip	Taip	Ne	Taip	Ne
Google meet	Taip	Taip	Taip	Ne	Taip
Microsoft Teams	Taip	Taip	Taip	Taip	Taip
Slack	Taip	Taip	Taip	Taip	Taip
Talkie (mūsų sistema)	Taip	Taip	Taip	Ne	Taip
Sistema/Funkcija	Kripto valiutų stebėjimas	Užrašinė	Orų stebėjimas	Naujienos	Nustatymai
Discord	Ne	Ne	Ne	Ne	Taip
Google meet	Ne	Ne	Ne	Ne	Taip
Microsoft Teams	Ne	Taip	Taip	Taip	Taip
Slack	Ne	Ne	Ne	Ne	Taip
Talkie (mūsų sistema)	Taip	Taip	Taip	Taip	Taip
Sistema		Taškai (Kiek surinko TAIP)			
Discord		4			
Google meet		5			
Microsoft Teams		9			
Slack		6			
Talkie (mūsų sistema)		9			

Sistema	Privalumai	Trūkumai
Discord	<ul style="list-style-type: none"> Paprasta ir lengvai suprantama vartotojo sąsaja. Didelis vartotojų kiekis. Daugybė bendruomenių turinčių savo kanalus platformoje. 	<ul style="list-style-type: none"> Specifinių bendruomenių kanalų reikia ieškoti kituose forumuose ar internete. Vaizdo transliavimo kokybė prasta.
Google Meet	<ul style="list-style-type: none"> Labai lengva naudotis Viską gali pasiekti keliais mygtukų paspaudimais. Galima planuoti tolimesnius susitikimus 	<ul style="list-style-type: none"> Būtina turėti Google paskyrą
Slack	<ul style="list-style-type: none"> Patogu bendrauti su dideliu kiekiu žmonių Lengva sukurti pokalbius ar kanalus Patogu stebėti savo kanalo analitiką 	<ul style="list-style-type: none"> Sudėtinga sąsaja naujam vartotojui Mažai papildomų įrankių palengvinančių darbą
Microsoft Teams	<ul style="list-style-type: none"> Lengva suskurti savo komandą ir į ją pakviesti kitus vartotojus Lengva pokalbio valdymo sistema Failų saugojimas Patogi susirašinėjimo su atskirais vartotojais sistema 	<ul style="list-style-type: none"> Sudėtinga vartotojo sąsaja. Nėra intuityvus sistemos naudojimas Suteikia daug papildomų įrankių tačiau kartu tai apsunkina sistemos naudojimą.

2. Sistemos teorinė dalis

Šiame skyriuje yra pateikiama teorinė informacija apie mūsų sukurtą sistemos modelį – sistemos funkcionalumas, funkciniai reikalavimai, aktoriai ir objektai, UML diagramos ir duomenų bazės modelis. Visa ši informacija bus pateikta toliau esančiuose poskyriuose.

2.1. Sistemos funkcionalumas

Kurdami šią sistemą mes susidarėme planą kaip apytiksliai mūsų sistema turėtų funkcionuoti, kiekvieną funkcionalumą bandėme apsirašyti.

- Vartotojų registracija į sistemą.
- Vartotojų prisijungimas prie sistemos.
- Vartotojų nukreipimas prisijungus į pradinį puslapį.
- Vartotojų galimybė redaguoti savo profilį.
- Vartotojo galimybė keisti savo slaptažodį
- Prisijungusių vartotojų galimybė sukurti vaizdo bei garso skambučius.
- Prisijungusių vartotojų galimybė prisijungti prie kitų vartotojų sukurtų vaizdo bei garso skambučių.
- Vartotojų galimybė naudotis kalendoriaus įrankiu, pridėti datas, ištrinti datas, planuoti susitikimus bei kitus įvykius.
- Vartotojų galimybė naudotis kripto valiutų įrankiu, stebėti kripto valiutas, prisidėti mėgstamas valiutas, panaikinti mėgstamas valiutas iš mėgstamų sąrašo.
- Vartotojų galimybė naudotis „To Do“ įrankiu, užsirašyti ir išsaugoti norimus darbus, ištrinti atliktus darbus.
- Vartotojų galimybė stebėti norimo miesto orus, išsaugoti savo mėgstamą miestą, pakeisti mėgstamą miestą.
- Vartotojų galimybė naudotis užrašinės įrankiu, pridėti užrašus, trinti užrašus.
- Vartotojų galimybė naudotis naujienų įrankiu, kuriame vartotojas gali pasirinkti norimas stebėti naujienas, jas keisti.

2.2. Aktoriai ir objektai

Mūsų sistemoje yra trys skirtingi vartotojų tipai, tai yra: neprisijungę vartotojai (anonimai), prisijungę vartotojai, kitaip tariant užsiregistravę sistemos vartotojai, sistemos moderatoriai ir administratoriai. Asmuo turintis tam tikrą tipą gali pasiekti skirtingą sistemos dalį ir funkcionalumą:

- **Anoniminiai vartotojai** gali pasiekti ir atlikti šias sistemos funkcijas:
 - Užsiregistruoti į sistemą.
- **Prisijungę vartotojai** gali pasiekti ir atlikti šias sistemos funkcijas:
 - Turi visas **Anoniminio vartotojo** funkcijas.
 - Prisijungti prie sistemos.
 - Turėti vaizdo skambučius.
 - Turėti balso skambučius.
 - Stebėti krypto valiutas, pridėti bei ištrinti jas iš stebimų sąrašo.
 - Pildyti darbus To Do įrankyje bei atliktus darbus ištrinti.
 - Stebėti orus pasirinktame mieste.
 - Pildyti kalendorių, ištrinti jau praėjusias datas.
 - Naudotis užrašine, pridėti ir trinti užrašus.
 - Stebėti norimas naujienas.
 - Redaguoti savo informaciją.
- **Sistemos moderatoriai** gali pasiekti ir atlikti šias sistemos funkcijas:
 - Turi visas **Prisijungusio vartotojo** funkcijas.
 - Stebėti įvairią su sistema susijusią statistiką
 - Stebėti ir redaguoti kitų vartotojų įrankių informaciją.
- **Sistemos administratoriai** gali pasiekti ir atlikti šias sistemos funkcijas:
 - Turi visas **Sistemos moderatoriaus** funkcijas.
 - Stebėti ir redaguoti kitų vartotojų profilio informaciją.
 - Ištrinti vartotojus.

Sistemoje yra 8 objektai, kurie turi tam tikras būsenas, kurios gali keistis priklausomai nuo vartotojo atliekamų veiksmų:

- **Kripto stebėjimo įrankis** – sukurti, ištrinti, stebėti.
- **„To Do“ įrankis** – sukurti, redaguoti, ištrinti.
- **Orų stebėjimo įrankis** – stebėti, pasirinkti.
- **Užrašinė** – sukurti, redaguoti, ištrinti.

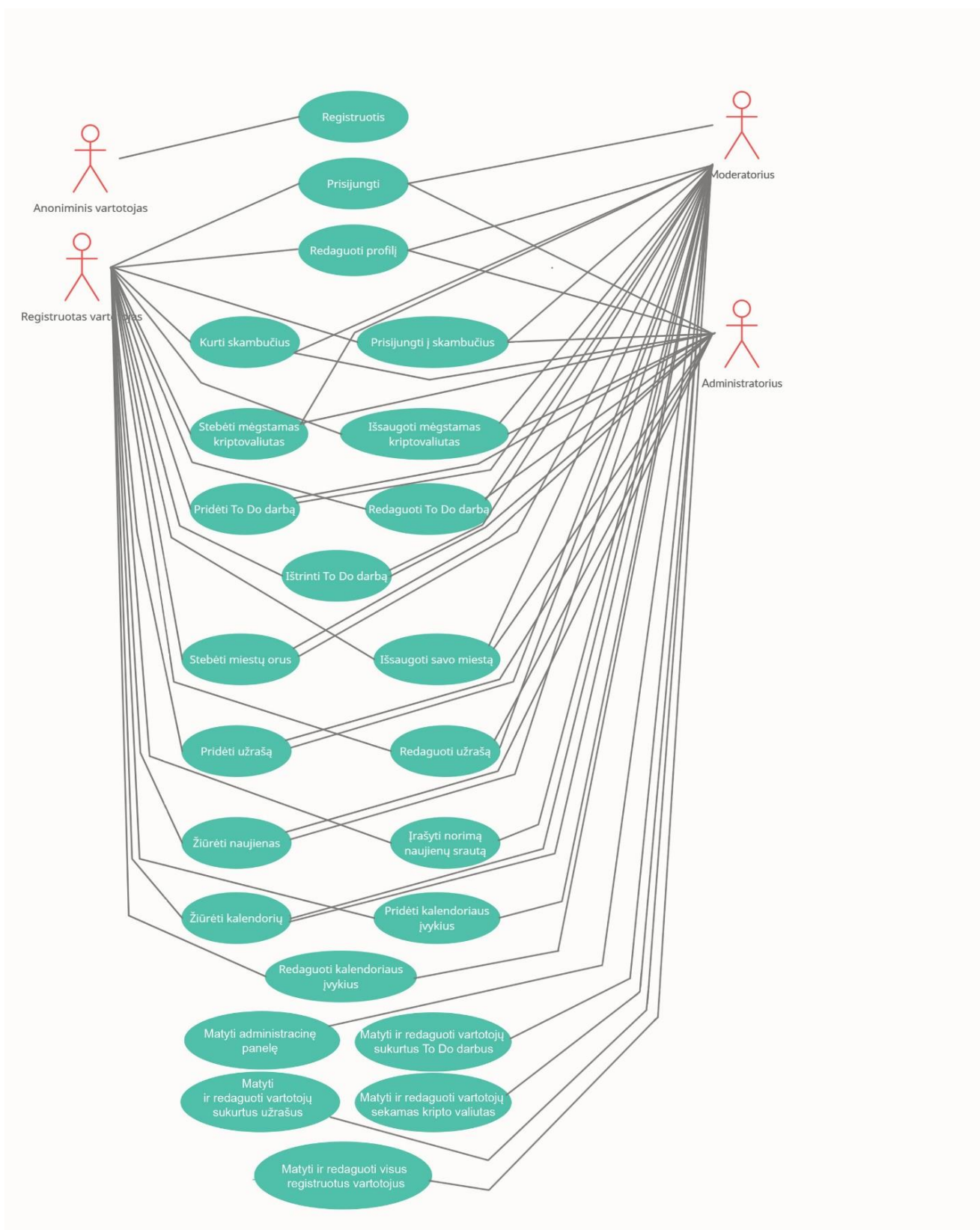
- **Naujienos** – stebėti, pasirinkti.
- **Kalendorius** – sukurti, redaguoti, ištrinti.
- **Skambutis** – sukurti.
- **Vartotojų duomenys** – sukurti, redaguoti, ištrinti.

2.3. UML diagramos

Šiame poskyryje yra pateikiamos mūsų sistemos funkcionalumo diagramos bei jų aprašymai. UML diagramos – naudojimo atvejų, klasių diagramos.

2.3.1. Naudojimo atvejų diagrama

Prieš tai buvusiame skyriuje detaliai aprašėme visus tipus bei kiekvieno tipo galimybes bei funkcionalumą. 1 pav. galite matyti vizualiai pavaizduotą visų tipų (aktorių) funkcionalumą. Lengvai galima pastebėti, jog aukštesnių teisių vartotojai paveldi visų, žemesnių, vartotojų teises ir gali naudotis tomis pačiomis funkcijomis ir įrankiais. Tačiau, sistemos saugumui užtikrinti, sistemos moderatoriai ir administratoriai gali stebėti ir valdyti vartotojų įkeliamą ir išsaugomą informaciją.

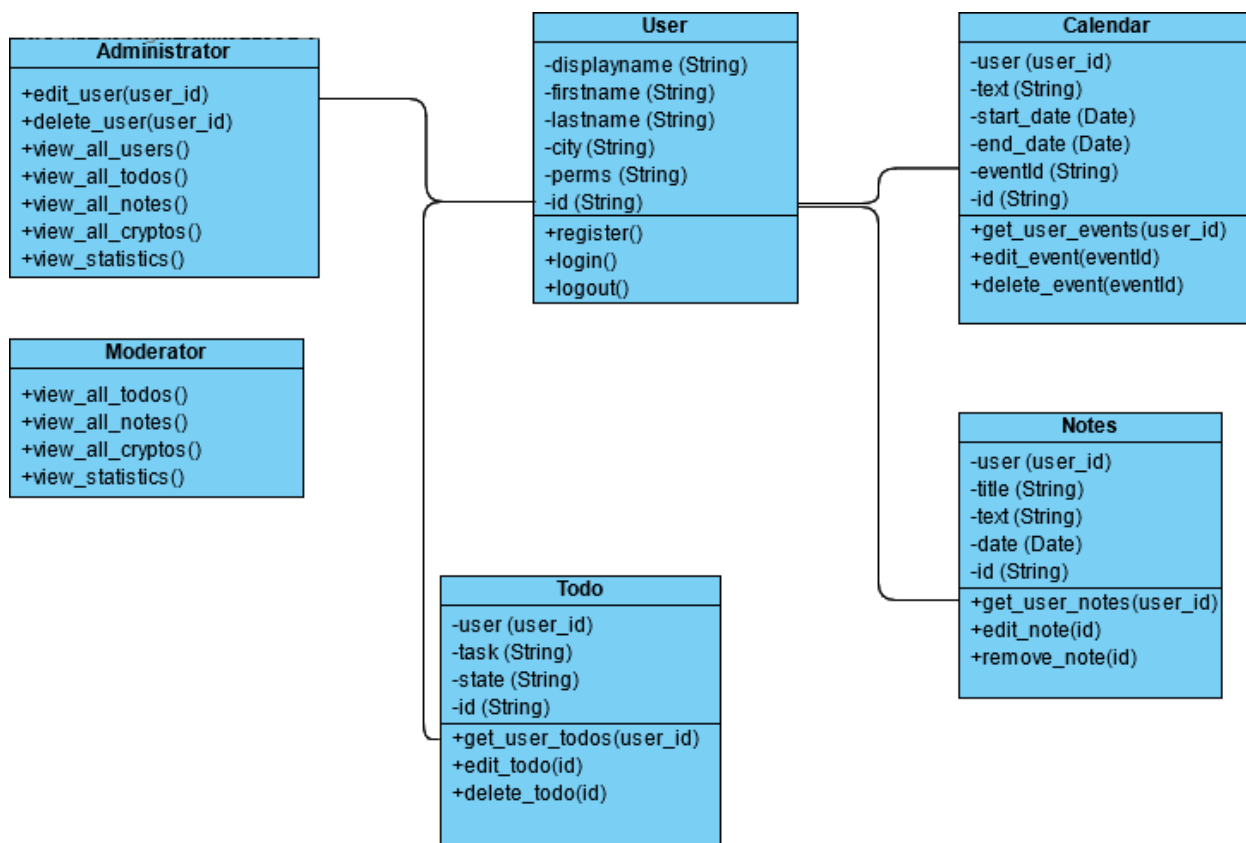


1 pav. Naudojimo atvejų diagrama

2.3.2. Klasių diagrama

Nors JavaScript yra funkcionali, o ne objektams orientuota kalba ir mūsų kode klasių ir klasių metodai neegzistuoja 2 pav. yra pavaizduota, ką gali atlikti kiekvienas vartotojas ir kokią funkcionalumą atlieka skirtingos kodo dalys.

pav. 2 yra pateikta sistemos klasių diagrama. Diagramoje atvaizduojami trijų tipų registruoti vartotojai – Paprastas vartotojas, moderatorius ir administratorius. Visi šie vartotojai duomenų bazėje atrodo taip pat tik su 1 skirtingu – perms (angl. permissions) laukeliu, kuris nusako vartotojo teisių lygį (normal, moderator arba admin), todėl visi šie vartotojai įprasto naudojimo metu niekuo nesiskiria. Visos kitos klasės priklauso nuo objekto, sukūrusio vartotojo ir laiko informaciją apie vartotojo sukurtus To Do darbus, kalendoriaus įvykius, užrašus ir t.t. arba sekamas naujienas ir kripto valiutas. Visų šių lentelių duomenys susiejami su jais sukūrusio vartotojo ID, kad būtų lengva atskirti kuri informacija priklauso kam.



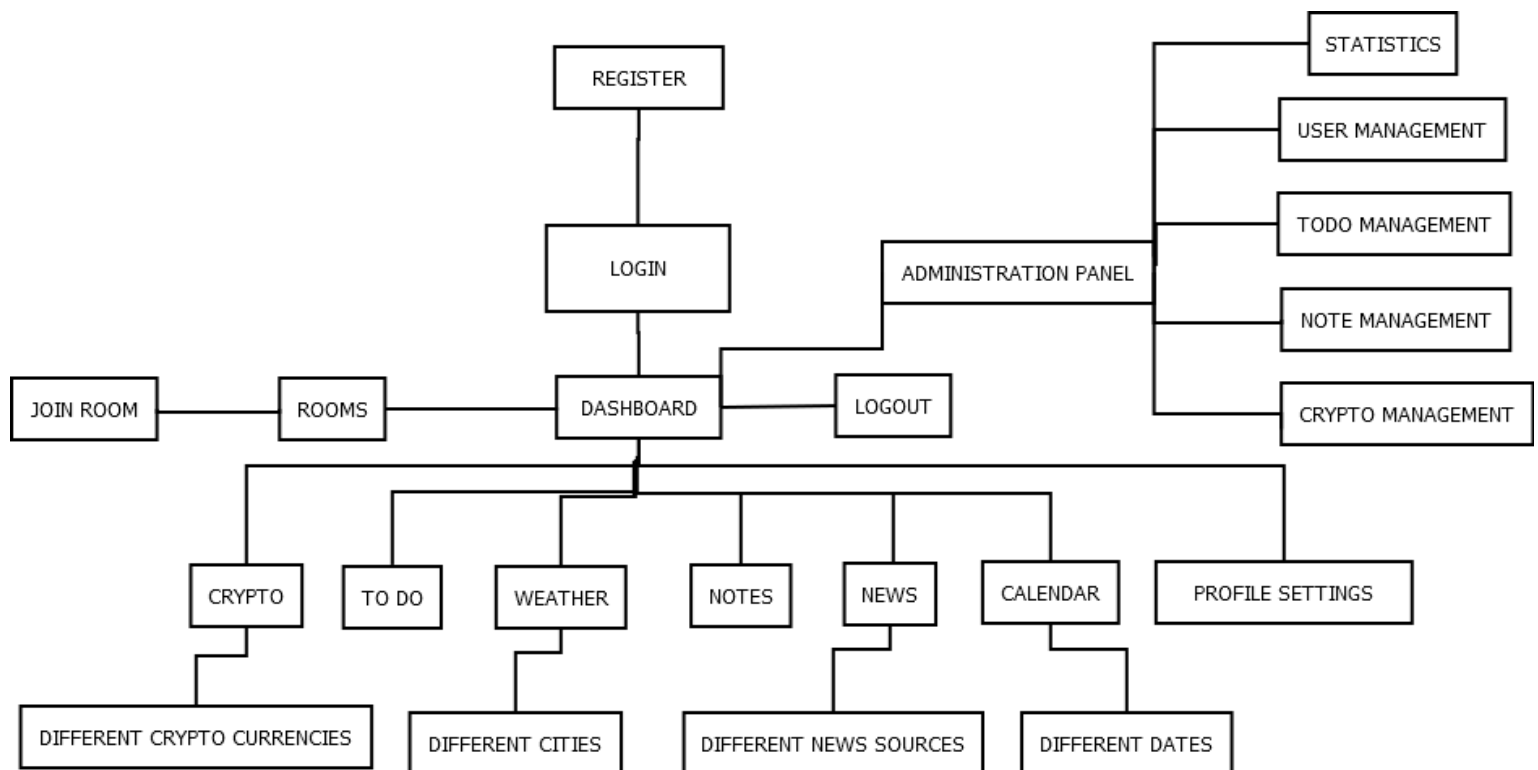
1 pav. Klasių diagrama

2.3.3. Sistemos navigacijos struktūra

Pradinis sistemos puslapis, kuris pateikiamas vartotojui pirmą kartą apsilankius puslapį – Login. Naudotojui norint toliau naudotis sistema reikia prisiregistruoti ar prisijungti. Norint prisiregistruoti yra sukurtas atskiras sistemos puslapis Register. Naudotojui prisiregistravus jis yra nukreipiamas atgal į Login puslapį, Vartotojui prisijungus jis yra nukreipiamas į pagrindinį puslapį (Dashboard), iš kurio vartotojas gali pasiekti kitus sistemos puslapius, kurie yra prieinami užsiregistravusiam vartotojui, kaip Crypto, To Do, Weather, Notes, News ir Calendar. Taip pat, vartotojas gali nueiti į Settings puslapį, kuriame gali keisti savo asmeninę informaciją. Iš Dashboard puslapio vartotojas gali sukurti ir pasiekti naują pokalbio kambarį, bei jau sukurtą kambarį.

Jeigu naudotojas turi teisę (moderatorius ar administratorius) gali pasiekti administracinį puslapį Admin, kuriame yra pateikiama bendra sistemos statistiką, bei turi galimybę peržiūrėti kitų vartotojų sukurtus įrašus. Jei naudotojo atžvilgiu įrašas yra kenkėjiškas ar kaip kitaip neteisingas, jis jį gali pakeisti arba ištrinti. Galiausiai, prisijungęs vartotojas, norėdamas atsijungti nuo sistemos gali paspausti mygtuką Logout, kuris yra pateikiamas tik prisijungusiems vartotojams.

Visa aprašyta sistemos navigacijos struktūra yra pateikiama 3 pav.



3 pav. Sistemos navigacijos struktūra

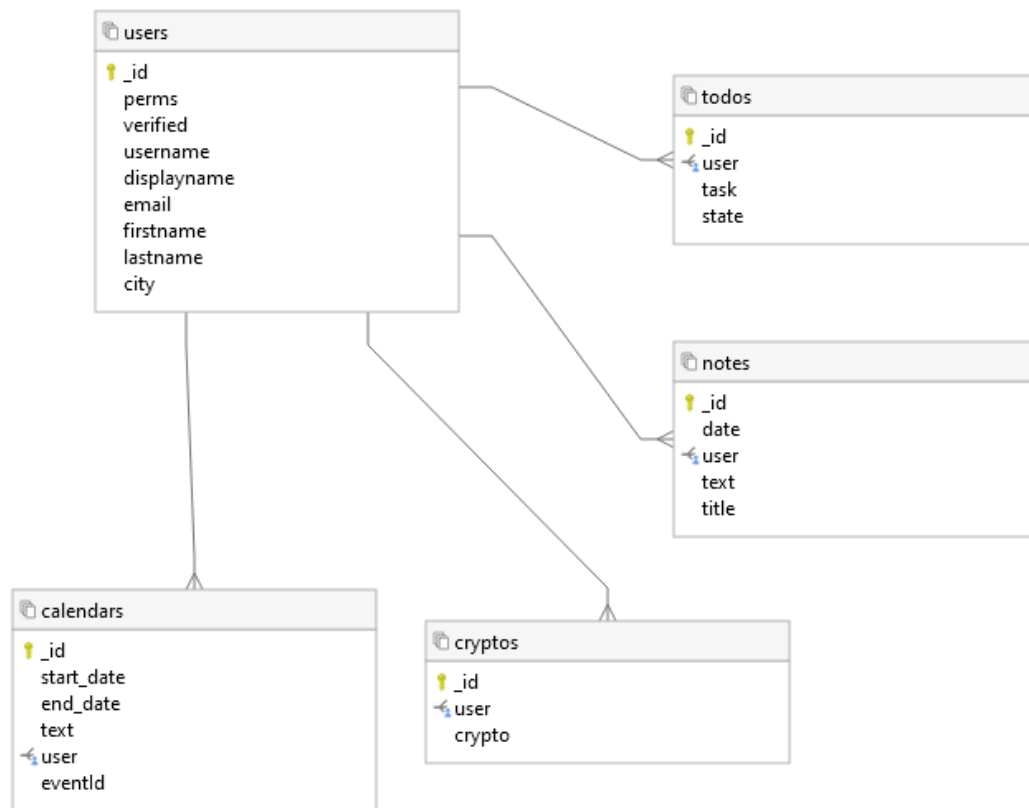
2.3.4. Duomenų bazės modelis

Sistemos duomenų saugojimui bei apdorojimui yra naudojama MongoDB duomenų bazių valdymo sistema, kuri turi platų duomenų tipų pasirinkimą, bei lengvą susiejimą su mūsų naudojama serverio sistema. Dar lengvesniam susiejimui serveryje naudojama biblioteka mongoose, kuri leidžia kurti, išrinti, redaguoti ir apdoroti duomenų bazės informaciją naudojant modelius. Visas šis funkcionalumas yra pagrįstas JavaScript objektais ir jis nereikalauja jokių tiesioginių NoSQL užklausų naudojimo.

Duomenų bazėje yra saugomos šios lentelės:

- **User** lentelėje yra saugoma informacija apie registruotus vartotojus. Registracijos metu vartotojas turi pateikti šiuos duomenis: el. paštą, slaptažodį, vardą ir pavardę. Taip pat, šioje lentelėje yra saugomi papildomi duomenys, kurie kinta vartotojui naudojantis sistema.
- **Todo** lentelėje yra saugomi vartotojų sukurti To Do darbai. Kuriant darbą vartotojas nurodo jam aprašymą ir sistema paskiria jam statusą „aktyvus“. Norint vartotojui, statusas gali būti pakeičiamas į „atliktas“, kuris nurodo, jog vartotojas šį To Do darbą jau atliko, arba jį ištrinti.
- **Notes** lentelėje yra saugomi vartotojo sukurti užrašai. Kiekvienas užrašas turi antraštę ir tekstą, kurie bet kuriuo metu gali būti redaguojami vartotojo. Taip pat išsaugant objektą lentelėje jam yra priskiriama sukūrimo data, kuri yra atvaizduojama vartotojui.
- **Crypto** lentelėje yra saugoma informacija apie vartotojo sekamas kripto valiutas. Vartotojui pasirinkus jį dominančias kripto valiutas, iki 8 valiutų yra išsaugomos lentelėje kaip masyvas. Vartotojas bet kuriuo metu gali keisti šį sąrašą, jei tik jis neviršija 8 iš viso sekamų kriptovaliutų skaičiaus.
- **Calendar** lentelėje yra saugoma informacija apie vartotojų sukurtus kalendoriaus įrašus. Kiekvienas įrašas turi pradžios ir pabaigos datą ir tekstą. Šie įrašai bet kuriuo metu gali būti keičiami vartotojo.

4 pav. yra pateikta ER diagrama, nurodanti duomenų bazės lenteles ir jų sąryšius



4 pav. duomenų bazės ER diagrama

3. Sistemos praktinis įgyvendinimas

Kuriama sistema daugiausiai yra susijusi su darbu prie kompiuterio, skirta kuo labiau sumažinti vartotojo naršymą po nereikalingus interneto portalus ar kitus įrankius. Dėl šios priežasties sistema yra labiausiai sutelkta į kompiuterio vartotojus, tačiau ja naudotis galima naudojant bet koki įrenginį. Kadangi sistema turi būti pasiekama iš visur, nes darbas bet kada gali iškviesti, mes pasirinkome sistemą daryti kaip internetinę aplikaciją, tai vienas iš paprasčiausių būdų lengvai pasiekti visus galimus vartotojus. Sistemai programuoti ir modeliuoti pasitelkėme ne vieną įrankį bei technologiją.

3.1. Naudojamos technologijos

Technologijos, kurias pasirinkome išorinės vartotojo sąsajos programavimui (front-end) yra: HTML – pateikti turiniui skirtam mūsų sistemai, CSS – atlikti įvairius su vizualizacija ir atvaizdavimu skirtus darbus, kitaip tariant apipavidalinti sistemą, JavaScript – interaktyviai sąsajai sistemoje kurti. Taip pat yra naudojamas Bootstrap karkasas, kuris padeda lengviau apipavidalinti sistemą, suteikia galimybę naudoti jau paruoštus apipavidalinius, svetainę daug lengviau padaryti dinamiška.

Vidinės sistemos pusės programavimui buvo pasitelktos šios technologijos: NodeJS – darbinė terpė (angl. Runtime environment) leidžia naudoti JavaScript kalbą kuriant serverius, MongoDB – NoSQL duomenų bazių valdymo sistema, lengvai integruojama į NodeJS serverio aplinką dėl panašaus duomenų saugojimo principo, Mongoose – NodeJS biblioteka, kuri leidžia dirbti su MongoDB kuriant modulius.

3.2. Sistemos funkcionalumas

Sistemos funkcionalumas susidaro iš mažesnių funkcijų ir maršrutų (angl. route), kuriems, vartotojui naudojantis sistema, siunčiamos įvairios užklausos ir iš jų gaunama informacija yra pateikiama naudotojui. Prieš kiekvieną užklausą informacija yra apdorojama įvairių tarpinių kodo dalių (angl. middleware) įsitikinančių, kad vartotojas turi teisę daryti šią užklausą, bei išvalančių (angl. sanitize) pateikiamą užklausą dėl saugumo.

Serverio dalis yra aprašyta ir naudojama tarsi REST API, su kuria bendrauja priekinė puslapio dalis (angl. front-end). Programinis kodas yra atskirtas į atskiras dalis, pagal tai, kokią funkciją jis atlieka, pavyzdžiui - kalendoriaus funkcionalumas yra išskirstytas į tris calendar.js failus, vienas failas nurodo

duomenų bazės modelį, sekantis failas atsakingas už teisingo failo pateikimą naudotojui kreipiantis į /calendar maršrutą, o paskutinis failas atlieka REST API funkcionalumą ir apdoroja bei išsaugo vartotojo pateiktus duomenis. Taip pat, gali būti dar vienas failas, kuriame laikomas JavaScript kodas suteikiantis funkcionalumą bendravimui su serverio dalimi kliento pusėje.

3.3. Problemos

Šio projekto metu dirbdami kartu susidūrėme su keliomis problemomis, kurios apsunkino mūsų projekto sistemos sėkmingą vystymą.

Viena iš pirmųjų problemų buvo skirtingas žinių kiekis interneto aplikacijų vystyme, tačiau šią problemą ganėtinai greitai išsprendėme, pasiskirstydami darbus taip, kad abiem būtų patogu dirbti vienas su kitu, kad keliai susikirstų tik suplanuotu laiku ir kad vienam nereiktų laukti kito atlikto darbo.

Viena iš pagrindinių problemų buvo naršyklių skirtumai - Chrome nepalaiko vaizdo pokalbio kameromis, dalinai išsprendėme šią problemą naudodami tik mikrofoną, jei kamera yra nepalaikoma.

Antroji problema su kuria susidūrėme buvo prisitaikantis puslapis (angl. Responsive page), kitaip tariant, kad puslapis tiktų kiekvienam ekrano dydžiui. Nors ir naudojome Bootstrap, kuris garsėja savo prisitaikymu, tačiau prieš tai nebuvo jo naudoję ir tai apsunkino vėlesnį puslapio tobulinimą, tačiau šią problemą išsprendėme naudodami media queries CSS parametą.

Trečioji problema buvo su mūsų naudojama šablonų karkasu Handlebars. Kadangi puslapiai yra išskirstyti į skirtingus šablonus, dažnai norint pridėti į puslapį Javascript kodą, jis prisidėdavo puslapio viduryje, kas sukdavo daug problemų su kitomis bibliotekomis ir kodo konteksto neturėjimo ar nesuvokimo. Šią problemą išsprendėme pagrindiniam šablonui pačioje apačioje pridėdami dalį šablono, kurią gali pakeisti kiti įkraunami šablonai. Tuomet, įdedami failai visada pridėdami apačioje failo.

3.4. Programinis kodas

Dinaminis vartotojo paskyros nustatymų atnaujinimas. Vartotojo nustatymuose galima keisti įvairius laukus, kaip: vartotojo vardas, vardas, pavardė, miestas, kurie visi turi skirtingas reikšmes duomenų bazėje, tam, kad nereiktų mygtukams kurti atskirų paspaudimo klausytojų (angl. event listener), mygtukams buvo pridėta papildoma informacija, pagal kurią dinamiškai yra dinamiškai sugeneruojamos klausytojų funkcijos, kurios atlieka visą funkcionalumą, kaip naujos reikšmės gavimas ir išsiuntimas į serverio dalį. Žemiau yra pateikta šio kodo nuotrauka 5 pav.

```
[...document.getElementsByClassName("settings-edit")].forEach(button => {
  button.addEventListener("click", async (e) => {
    const fieldKey = button.getAttribute("attr-field");
    const fieldPretty = button.getAttribute("attr-pretty");
    const fieldCurrent = button.getAttribute("attr-current");
    const { value: newValue } = await Swal.fire({
      title: `Set new ${fieldPretty}`,
      input: 'text',
      inputPlaceholder: `${fieldCurrent}`
    });

    if (!newValue || (newValue == fieldCurrent)) {
      return;
    }
    const options = {};
    options[fieldKey] = newValue;
    const response = await postData("/api/user/", options, "PATCH");
    if(response.success) {
      document.location.reload();
    }
  });
});
```

2 pav. Dinaminis vartotojo paskyros nustatymų atnaujinimas

3.5. Saugumas

Kadangi vartotojai beveik visuose puslapio dalyse gali išsaugoti ar redaguoti tekstą, reikėjo stipriai apsisaugoti nuo XSS (cross site scripting) ir NoSQL Injection atakų. Šis saugumas buvo pasiektas „valant“ (angl. sanitize) duomenys prieš juos išsaugant į duomenų bazę ir rodant vartotojui. Taip pat naudojamos įvairios bibliotekos padidinančios naudojamo karkaso saugumą. Kad vartotojų duomenys būtų saugūs atakos atveju, visi slaptažodžiai yra užšifruojami SHA256 algoritmu ir atsitiktiniu salt, todėl, net jei ir kažkas gautų duomenų bazės informaciją, slaptažodžių atkurti negalėtų.

Vartotojų registracijai ir prisijungimui naudojama populiari biblioteka PassportJS, tam, jog užtikrinti, kad vienoje svarbiausių sistemos dalių nėra spragų. Tam, kad pristabdyti piktavalius nuo įsilaužimo į jiems nepriklausančias paskyras, į vartotoją galima bandyti prisijungti tik 5 kartus per penkias minutes.

Išvados ir rekomendacijos

Šio kursinio darbo metu buvo siekiam sukurti sistemą, kuri vartotojams suteiktų galimybę naudotis tik viena svetaine pasiekti visus šiomis dienomis naudojamus įrankius, sutrumpinant vartotojo navigacija tarp kitų aplikacijų ar programų. Pagrindinis išsikeltas tikslas buvo suprogramuoti internetinę sistemą, kurioje vartotojai, galėtų lengvai naudotis svetainės suteiktais įrankiais (Naujienos, Orai, Kalendorius, Kripto kainų sekimas, Užrašinė, „To Do“ užrašinė), modifikuoti savo profilius, galėtų lengvai susisiekti su kitais vartotojais vaizdo arba garso skambučiais. Galutiniame rezultate yra sukurta internetinė svetainė, kurioje vartotojai turi galimybę:

- Užsiregistruoti ir prisijungti.
- Turėti vaizdo bei balso skambučius.
- Prisidėti į sąrašą ir stebėti savo mėgstamas kripto valiutas.
- Susidaryti, išsaugoti, ištrinti ar išbraukti darbus „To Do“ įrankyje.
- Stebėti pasirinktus Orus.
- Pildyti užrašus užrašinės įrankyje.
- Stebėti pasirinktas naujienas naujienų skiltyje.
- Pildyti ir stebėti kalendorių, žymėti įvykius ir datas.
- Redaguoti profilį.

Atsižvelgiant į viso sistemos realizavimo procesą bei gautus rezultatus pateikiamos tokios **išvados**:

- Atlikus sukurtų sprendimų analizę pastebėta, kad šiais laikais dažnai kuriamos bendravimo sistemos yra orientuotos į vieną pagrindinį funkcionalumą - tai yra bendravimą, tačiau nesuteikia vartotojams kitų įrankių, nesusijusių su pagrindiniu funkcionalumu.
- Pastebėjome, kad nagrinėtose sistemose yra labai populiari savyje turėti ir failų saugojimo bei perdavimo funkcijas.
- Planuodami šią sistemą pastebėjome, kad naudojant panašias, išanalizuotas sistemas, vartotojas didelę laiko dalį turi praleisti ieškodamas papildomų įrankių, nesusijusių ir neintegruotų į pradinę sistemą.
- Analizuodami panašias sistemas, pastebėjome, jog platų kitų funkcijų pasirinkimą suteikiančios sistemos, kurios stengiasi suteikti vartotojui galimybę rasti viską vienoje sistemoje, kartu turi žymiai sudėtingesnę vartotojo sąsają bei reikalauja gilesnio įsitraukimo ir nagrinėjimo, kaip naudotis tokia sistema.

Rekomendacijos tolimesniam darbo vystymui:

- Pirmas pasiūlymas būtų išplėsti funkcionalumą ir pridėti susirašinėjimų skiltį, kad galėtų bendrauti ne vien vaizdo ar balso skambučiais.
- Kita naudinga funkcija būtų vartotojams turėti visiems prieinamus profilius, panašiai kaip socialiniuose tinkluose, kad būtų lengviau bendrauti, susirašinėti ar kalbėti gyvai.
- Taip pat įdomus funkcionalumas būtų įvairesnis įrankių pasirinkimo būdas, vartotojas galėtų į savo profilį prisidėti tik jam naudojamus įrankius iš duoto sąrašo, juos galėtų visada redaguoti pats.
- Taip pat naudingas funkcionalumas būtų sistemos temos, išvaizdos pakeitimas.

Literatūros sąrašas

- [1] Prisma's Top 11 NodeJS ORMs and database libraries. <https://www.prisma.io/dataguide/database-tools/top-nodejs-orms-query-builders-and-database-libraries> Žiūrėta: 2021-03-25
- [2] Mozilla Express web framework. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs Žiūrėta: 2021-02-28
- [3] MongoDB will not prevent NoSQL injections. <https://blog.sqreen.com/mongodb-will-not-prevent-nosql-injections-in-your-node-js-app/> Žiūrėta: 2021-04-16
- [4] Passport-Local Mongoose. <https://www.npmjs.com/package/passport-local-mongoose> Žiūrėta: 2021-03-29
- [5] Everything you need to know about the passport local strategy. <https://levelup.gitconnected.com/everything-you-need-to-know-about-the-passport-local-passport-js-strategy-633bbab6195?gi=e202dce394bf> Žiūrėta: 2021-03-29
- [6] User Authentication with Passport.js. <https://mherman.org/blog/user-authentication-with-passport-dot-js/> Žiūrėta: 2021-04-03
- [7] Passport Local Options. <https://github.com/saintedlama/passport-local-mongoose#options> Žiūrėta: 2021-04-18