

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

T120B029 Programų sistemų analizės ir projektavimo įrankiai
Projekto ataskaita nr. 3

Tema: Nuomos klubas

Studentų komanda:

Vytenis Kriščiūnas

Nojus Stauskas

Tomas Jasulevičius

Dėstytojas:

Eduardas Bareiša

KAUNAS, 2024

Turinys

1	Kuriamos sistemos aprašymas	3
2	Sistemos naudotojo sąsajos eskizai.....	3
3	Sistemos reikalavimų specifikacija	4
3.1	Panaudojimo atvejų modelis	4
3.2	Dalykinės srities modelis.....	8
3.3	Pasirinkta programavimo kalba, technologija	8
4	Sistemos projekto modelis	9
4.1	Sistemos loginė architektūra	9
4.2	Panaudojimo atvejų sekų diagramos.....	10
4.3	Sistemos klasių diagrama.....	23
4.4	Sistemos prototipas	24
5	Sistemos realizacijos modelis	27
5.1	Komponentų diagrama	27
5.2	Diegimo diagrama.....	27
6	Sistemos realizacija	28
6.1	Sekų diagramos kodo realizavimas	28

1 Kuriamos sistemos aprašymas

Kuriama daiktų nuomojimo sistema, tai interneto puslapis, kuriame vartotojai turės galimybę įkelti skelbimus. Skelbimai bus skirstomi į 2 pagrindines kategorijas, tai automobiliai ir įrankiai. Norint naršyti ir peržiūrėti skelbimų sąrašus, prisijungti prie sistemos nereikia. Tačiau norint naudotis kitomis sistemos funkcijomis bus reikalinga prisijungti. Neprisijungusiam vartotojui sistema suteiks galimybę užsiregistruoti į sistemą, tai yra susikurti paskyrą. Prisijungęs vartotojas gali rezervuoti skelbimą, įkelti skelbimą, nuomoti daiktą, valdyti savo paskyrą bei valdyti savo įkeltus skelbimus. Vartotojas pasirinkęs nuomojamą prekę turi pasirinkti galimas ir norimas datas iš kalendoriaus. Atlikus rezervaciją yra nuskaitomas atitinkamas turimų asmens kreditų kiekis. Yra suteikiama galimybė grąžinti pinigus apie tai pranešus administratoriui. Rezervacija turi būti patvirtinta skelbimo autoriaus per nurodytą laiką, antraip bus atliekamas pinigų grąžinimas paslaugos pirkėjui. Taip pat egzistuos administratoriaus rolė, kuri bus atsakinga už skelbimų kategorijų valdymą, vartotojų valdymą bei sisteminių pranešimų valdymą.

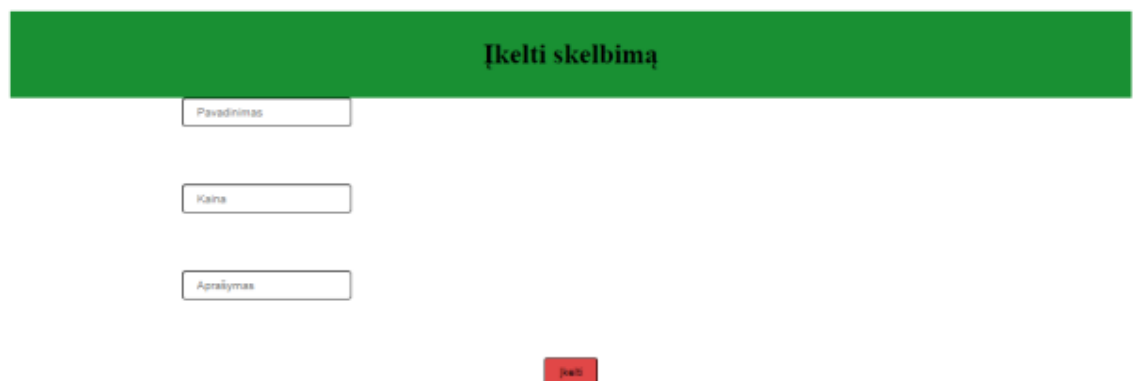
Naudotojai – tai vartotojai suinteresuoti nuomotis prekes susijusias su mašinomis ir jų taisymo įrankiais. Taikymo sritis – materialinė, siekiama užtikrinti veiksmingą sąryšį tarp paslaugos pirkėjo ir pardavėjo. Paslaugos pirkėjas – nuomojasi prekę, o nuomotojas gauna piniginį atlygį.

2 Sistemos naudotojo sąsajos eskizai



Eskizas pagrindinio puslapio. Viršutinė juosta yra žalia, su baltu tekstu: „Peržiūrėti sekamus skelbimus“, „Ieškoti profilio“, „Įkelti skelbimą“. Dešinėje yra raudoni mygtukai „Prisijungti“ ir „Registruotis“. Po juosta yra baltas laukelis su tekstu „Ieškoti skelbimo“.

1 pav. Pagrindinio puslapio eskizas

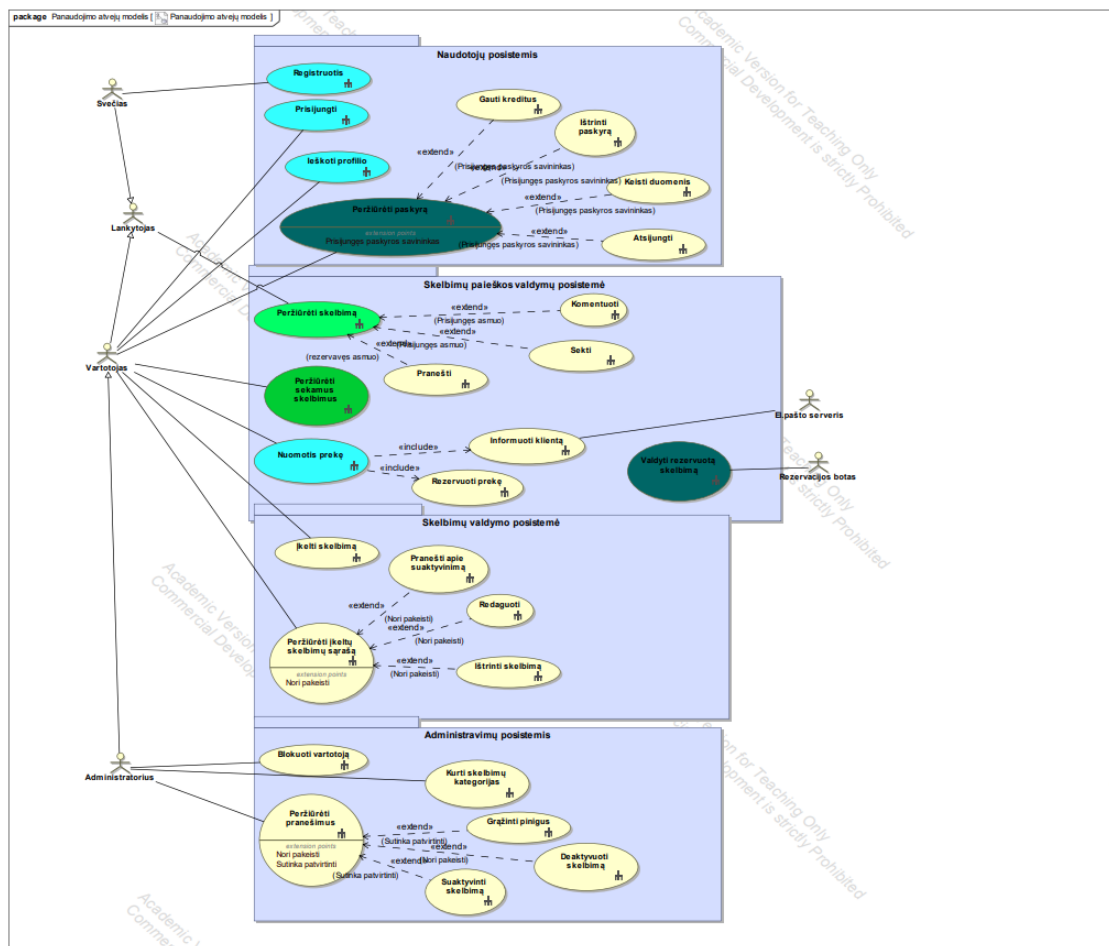


Eskizas skelbimo įkėlimo puslapio. Viršutinė juosta yra žalia, su baltu tekstu: „Įkelti skelbimą“. Po juosta yra trys balti laukeliai su tekstu: „Pavadinimas“, „Kaina“, „Aprašymas“. Po laukeliais yra raudonas mygtukas „Įkelti“.

2 pav. Skelbimo įkėlimo puslapio eskizas

3 Sistemos reikalavimų specifikacija

3.1 Panaudojimo atvejų modelis



3 pav. PA modelis

PA nr. 1 Registruotis		NS
Tikslas. Suteikti galimybę vartotojui prisiregistruoti prie sistemos		
Aprašymas. Neregistruotas vartotojas gali pasirinkti registracijos mygtuką, užpildyti duomenis ir užsiregistruoti į sistemą.		
Prieš sąlyga		Narys yra neprisijungęs
Aktorius		Svečias
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Narys gali prisijungti su registracijoje naudojamais duomenimis

PA nr. 2 Prisijungti		NS
Tikslas. Suteikti galimybę vartotojui prisijungti prie sistemos		
Aprašymas. Registruotas vartotojas gali pasirinkti prisijungimo mygtuką, užpildyti duomenis ir prisijungti į sistemą		
Prieš sąlyga		Neprisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Sistema atvaizduojama kaip prisijungusiam vartotojui

PA nr. 3 Ieškoti profilio		TJ
Tikslas. Suteikti galimybę vartotojui surasti kitų vartotojų paskyras		
Aprašymas. Vartotojas gali įvesti kito vartotojo vardą į paieškos laukelį ir peržiūrėti jo profilį iš pateikto sąrašo.		
Prieš sąlyga		Prisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Peržiūrėtas kito vartotojo profilis

PA nr. 4 Peržiūrėti paskyrą		TJ
Tikslas. Peržiūrėti savo paskyros duomenis		
Aprašymas. Vartotojas gali peržiūrėti savo paskyrą, jam prireikus jis gali keisti paskyros duomenis, ištrinti paskyrą ar atsijungti.		
Prieš sąlyga		Prisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	Ištrinti paskyrą, Keisti duomenis, Atsijungti
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Pakeisti/nekeisti paskyros duomenys arba atsijungta iš sistemos

PA nr. 5 Peržiūrėti skelbimą		VK
Tikslas. Suteikti galimybę peržiūrėti skelbimą lankytojiui		
Aprašymas. Lankytojas įvedęs į paiešką randa skelbimus ir gali pasirinkti peržiūrėti juos.		
Prieš sąlyga		
Aktorius		Lankytojas
Susiję panaudojimo atvejai	Išplečiantys PA	Komentuoti, Sekti
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Peržiūrėtas skelbimas

PA nr. 6 Peržiūrėti sekamus skelbimus		VK
Tikslas. Suteikti galimybę vartotojui peržiūrėti sekamus skelbimus		
Aprašymas. Vartotojas pasirinkęs peržiūrėti sekamus skelbimus mato sąrašą skelbimų kuriuos jis seka .		
Prieš sąlyga		Prisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Peržiūrėti sekami skelbimai

PA nr. 7 Nuomotis prekę		NS
Tikslas. Suteikti vartotojui galimybę išsinuomoti prekę		
Aprašymas. Vartotojas esantis skelbimo peržiūroje gali pasirinkti nuomotis prekę. Tuomet jis pasirenka laisvas datas kuriomis nori išsinuomoti šią prekę ir susimoka už ją.		
Prieš sąlyga		Prisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	Pranešti
	Apimami PA	Informuoti klientą, rezervuoti prekę
	Specializuojami PA	
Po sąlyga		Prekę užrezervuojama

PA nr. 8 Įkelti skelbimą		VK
Tikslas. Patalpinti į sistemą skelbimo įrašą.		
Aprašymas. Vartotojas gali patalpinti į sistemą skelbimą savo prekės išnuomavimui.		
Prieš sąlyga		Prisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Skelbimas patalpintas sistemoje

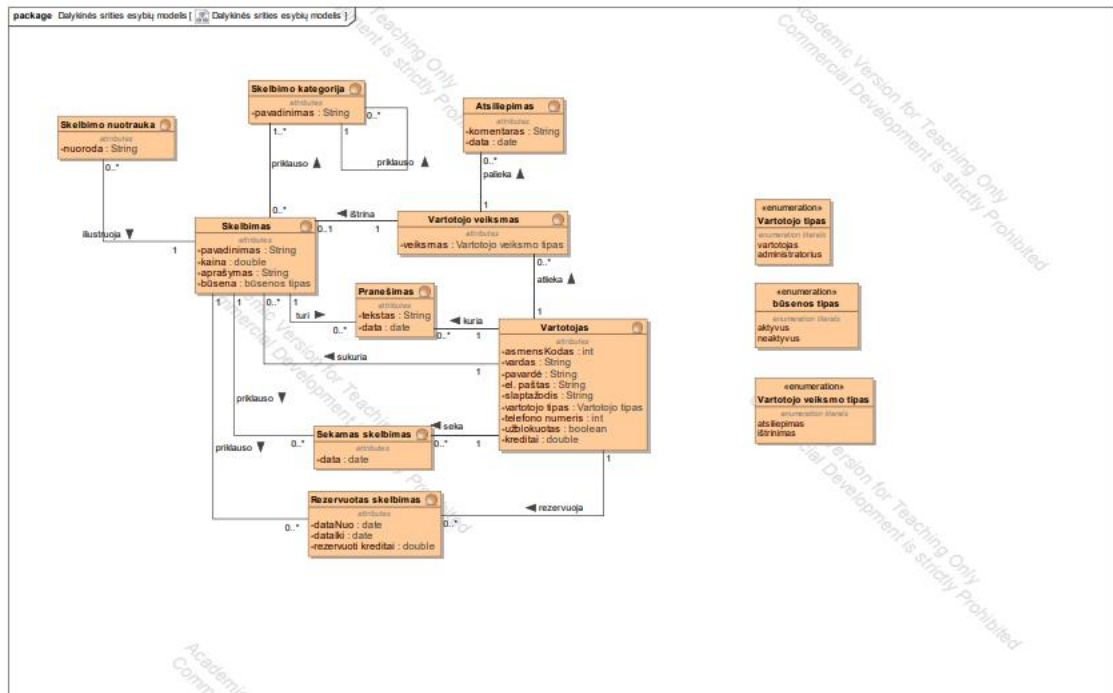
PA nr. 9 Peržiūrėti įkeltų skelbimų sąrašą		TJ
Tikslas. Suteikti vartotojui galimybę peržiūrėti jo paties įkeltus skelbimus		
Aprašymas. Vartotojas pasirinkęs peržiūrėti įkeltų skelbimų sąrašą mato visus skelbimus, kuriuos jis yra įkėlęs.		
Prieš sąlyga		Prisijungęs vartotojas
Aktorius		Vartotojas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Peržiūrėti įkelti skelbimai

PA nr. 10 Blokuoti vartotoją		TJ
Tikslas. Atriboti vartotojo galimybę vartoti sistema		
Aprašymas. Administratorius gali užblokuoti vartotoją, atimant iš jo galimybę sąveikauti su sistema		
Prieš sąlyga		Prisijungęs administratorius
Aktorius		Administratorius
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Vartotojas užblokuotas

PA nr. 11 Peržiūrėti pranešimus		NS
Tikslas. Suteikti galimybę administratoriui peržiūrėti pranešimus apie skelbimus		
Aprašymas. Administratorius gali peržiūrėti pranešimus apie skelbimus iš vartotojų ir esant reikalui grąžinti sumokėtus pinigus už nuomą.		
Prieš sąlyga		Prisijungęs administratorius
Aktorius		Administratorius
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Peržiūrėti pranešimai

PA nr. 12 Valdyti skelbimų būsenas		VK
Tikslas. Nustatyti skelbimą kaip aktyvų ar neaktyvų		
Aprašymas. Botas kas tam tikrą laiko intervalą tikrina ar sėkmingai buvo patvirtinta skelbimo nuoma, jei ne, botas padaro skelbimą neaktyviu ir grąžina pinigus.		
Prieš sąlyga		
Aktorius		Rezervacijos botas
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuojami PA	
Po sąlyga		Būsena suvaldyta

3.2 Dalykinės srities modelis



4 pav. DS modelis

Esybės „Vartotojas“ atributas kreditai – tai mūsų sistemoje naudojama valiuta, skirta atlikti transakcijoms.

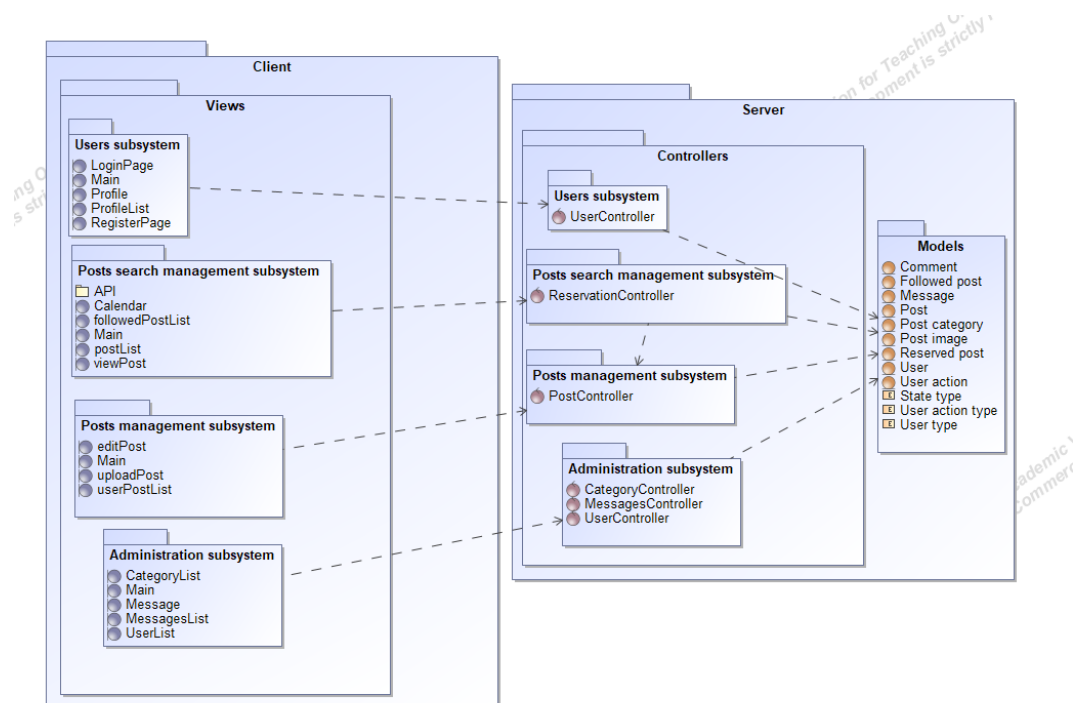
Esybės „Skelbimas“ atributas būsena – tai indikatorius, suteikiantis sistemos vartotojams galimybę rezervuoti skelbimą, jei būsena yra aktyvi.

3.3 Pasirinkta programavimo kalba, technologija

Pasirinkome naudoti React ir Node.js kartu su MySQL duomenų baze dėl jų galingų galimybių, didelio bendruomenės palaikymo ir lankstumo. React yra populiari biblioteka, kuri leidžia kurti modernius ir efektyvius vartotojo sąsajos komponentus, o Node.js yra lengvai išplėčiama ir efektyvi serverio pusės platforma. MySQL duomenų bazė buvo pasirinkta dėl jos patikimumo, greičio ir plačiai pripažintos naudojimo įmonių aplinkose.

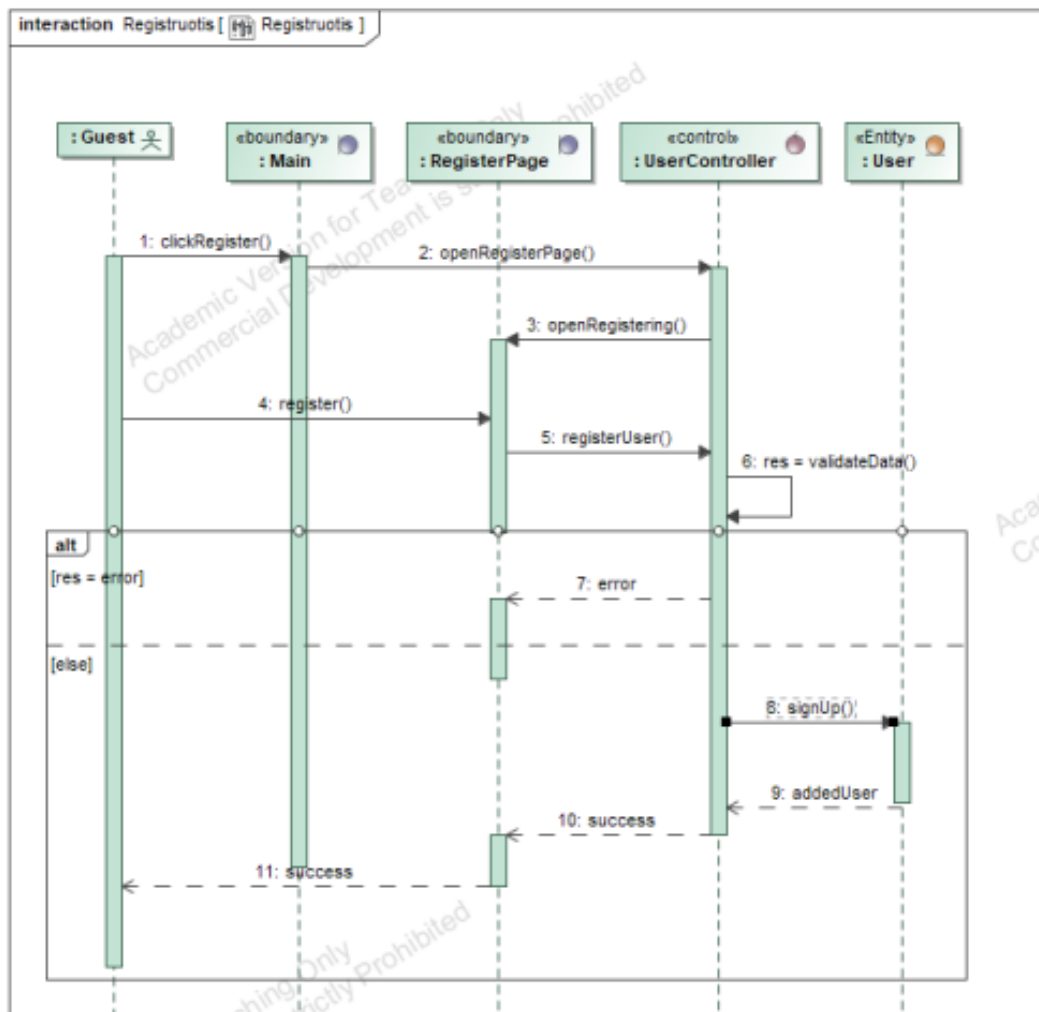
4 Sistemos projekto modelis

4.1 Sistemos loginė architektūra

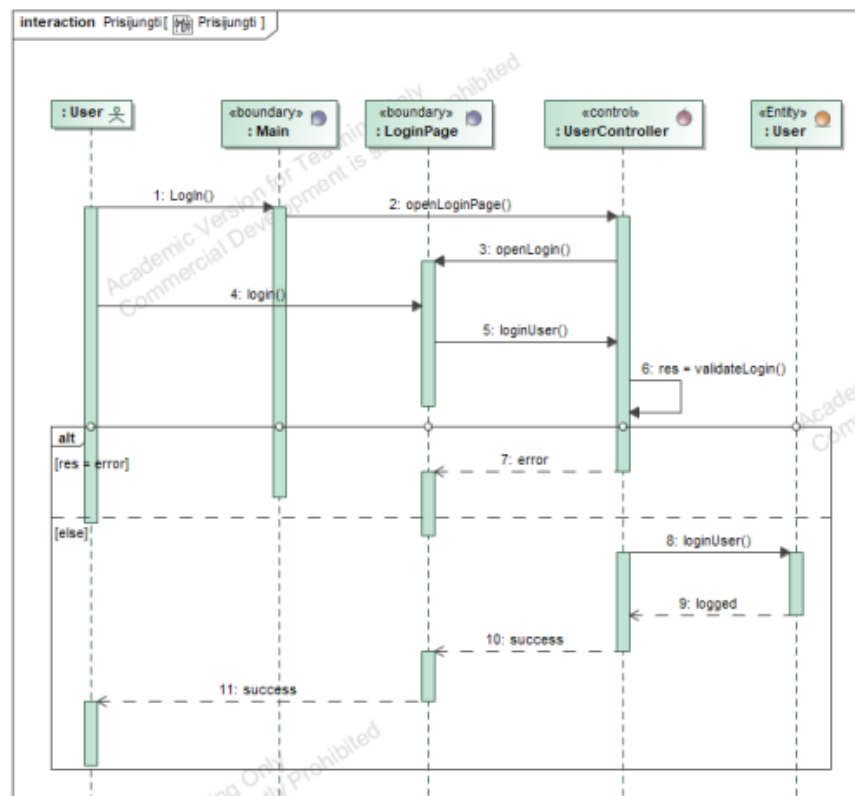


5 pav. Loginės architektūros schema

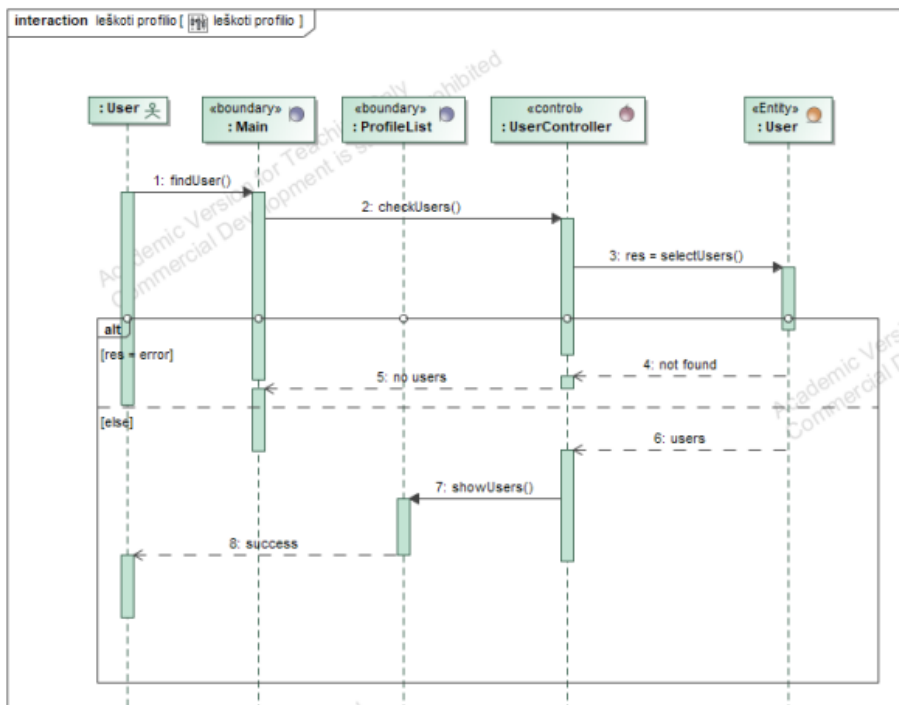
4.2 Panaudojimo atvejų sekų diagramos



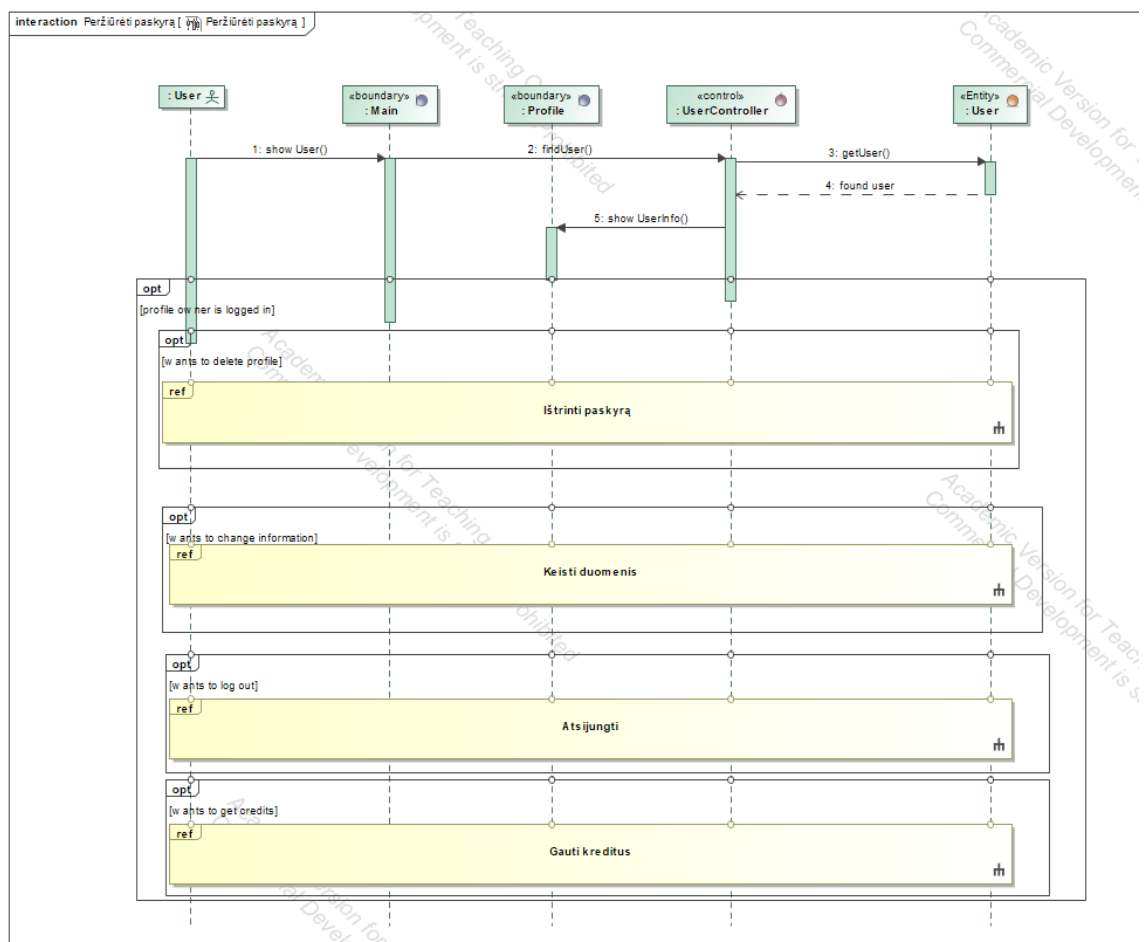
6 pav. Registruotis sekų diagrama



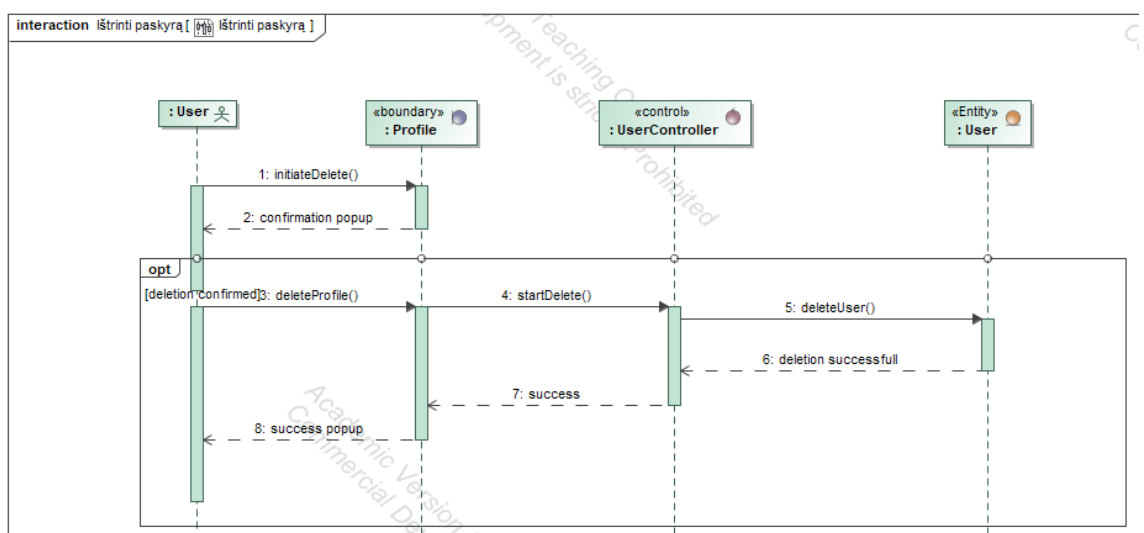
7 pav. Prisijungti sekų diagrama



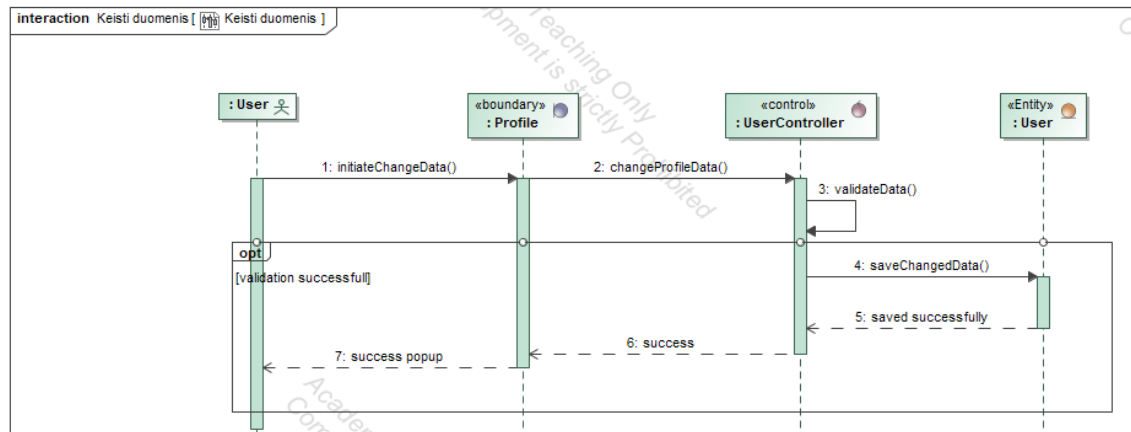
8 pav. Ieškoti profilio sekų diagrama



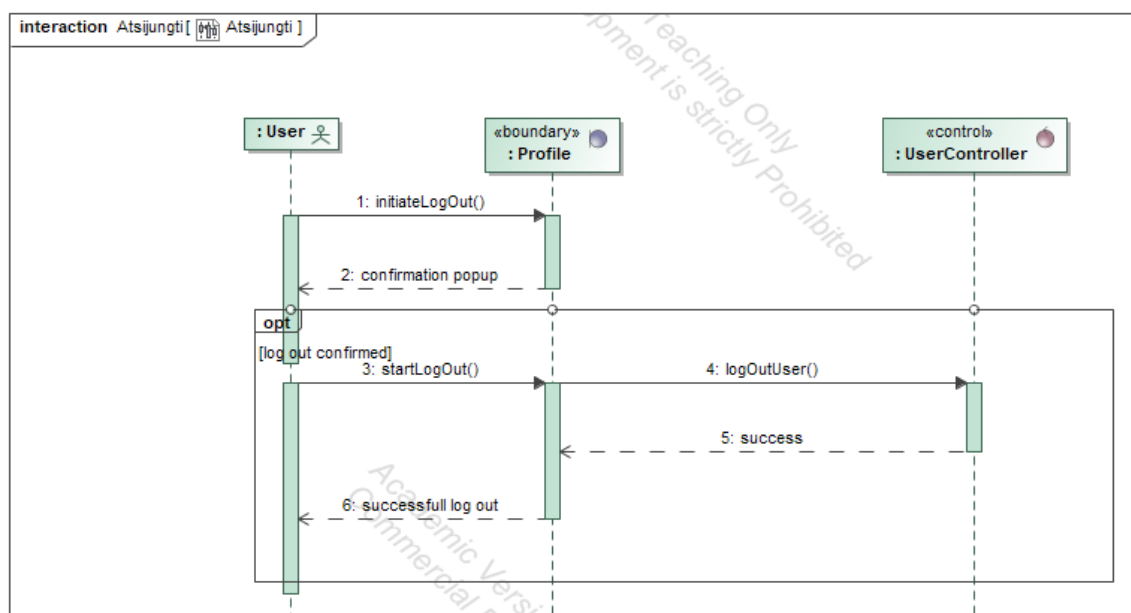
9 pav. Peržiūrėti paskyrą sekų diagrama



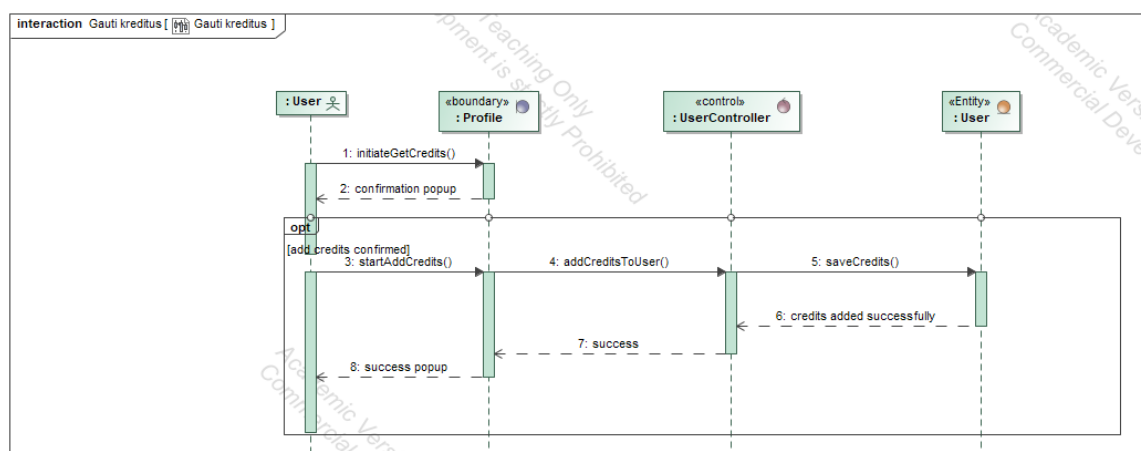
10 pav. Ištrinti paskyrą sekų diagrama



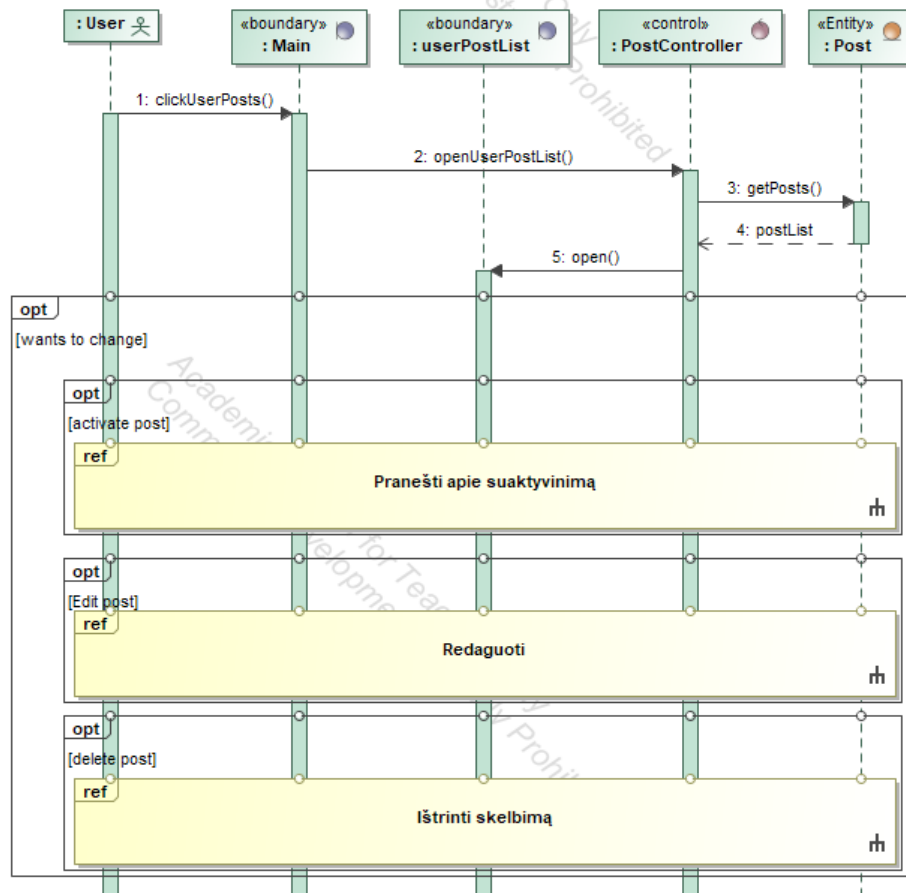
11 pav. Keisti paskyros duomenis sekų diagrama



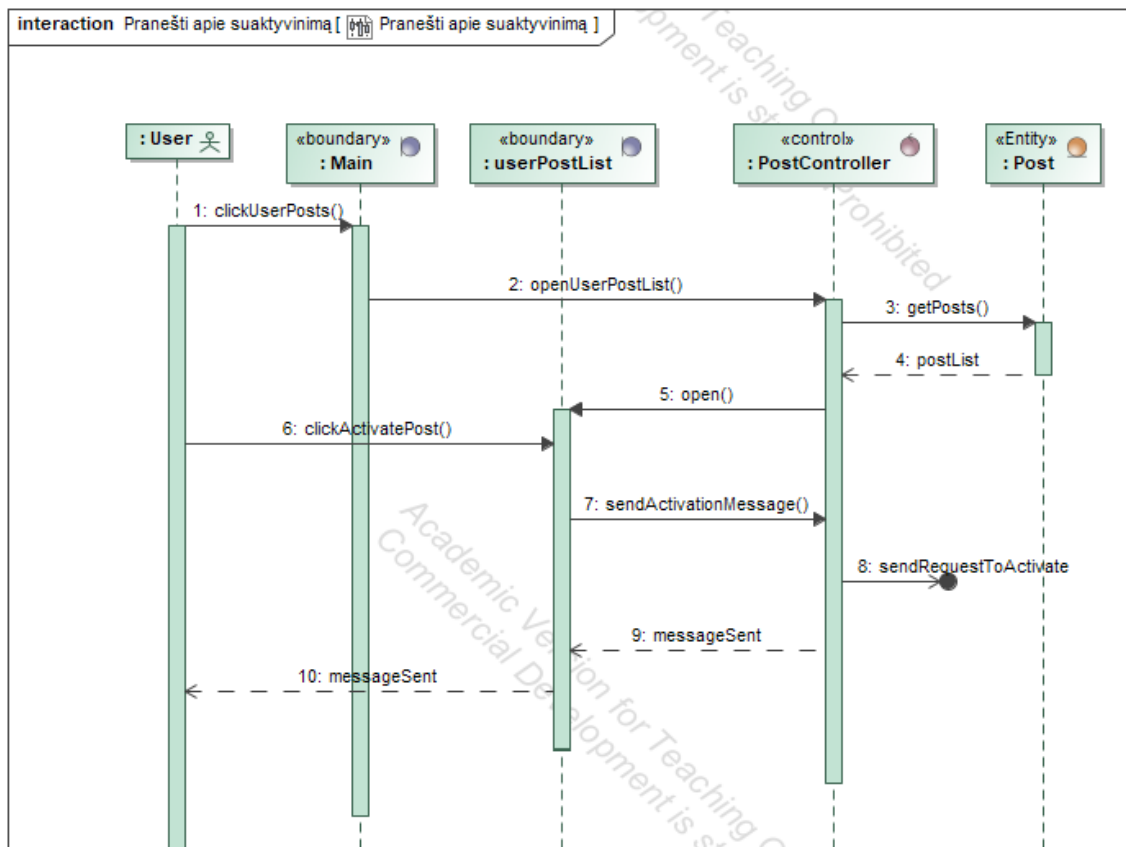
12 pav. Atsijungti sekų diagrama



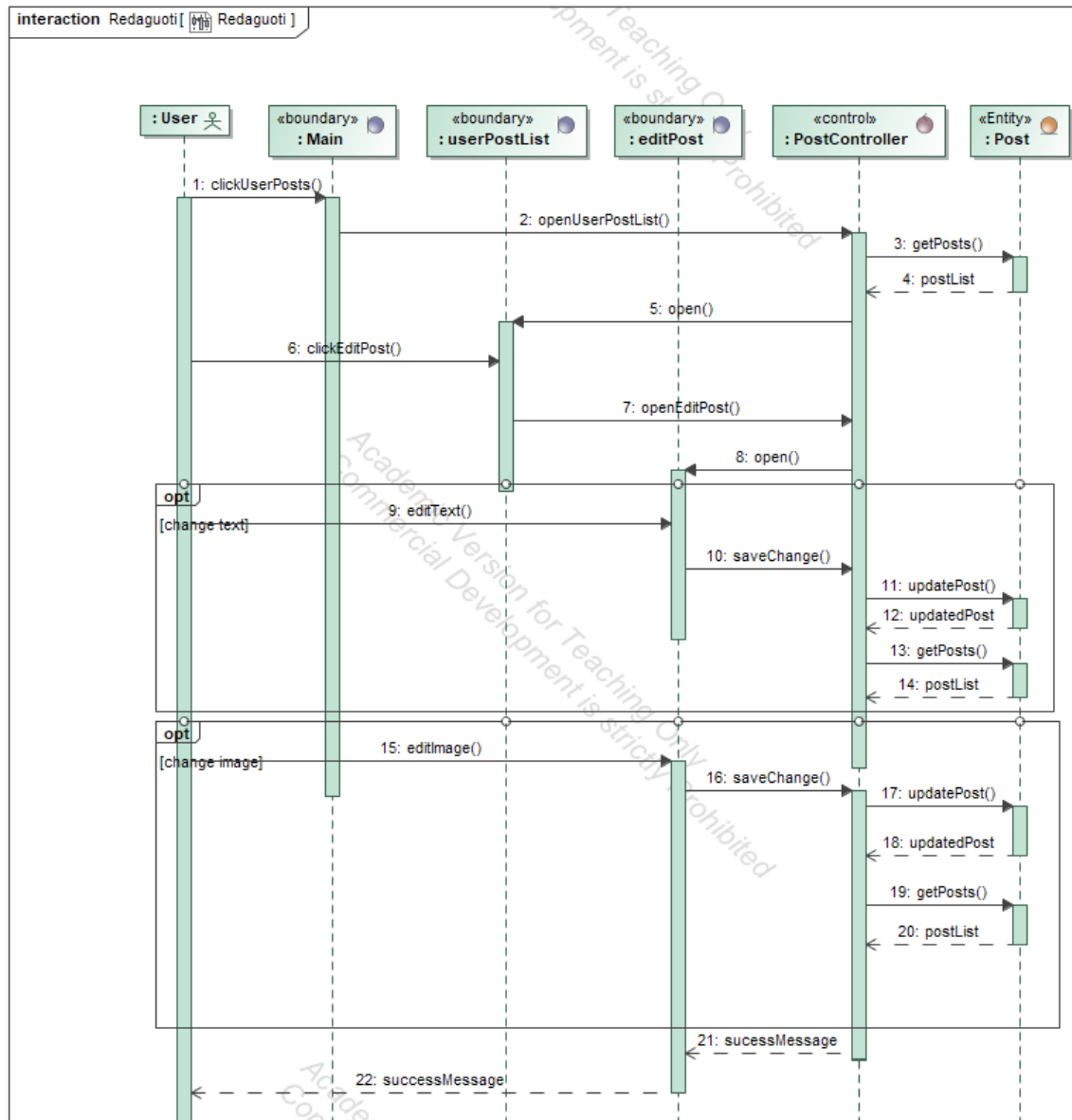
13 pav. Gauti kreditus sekų diagrama



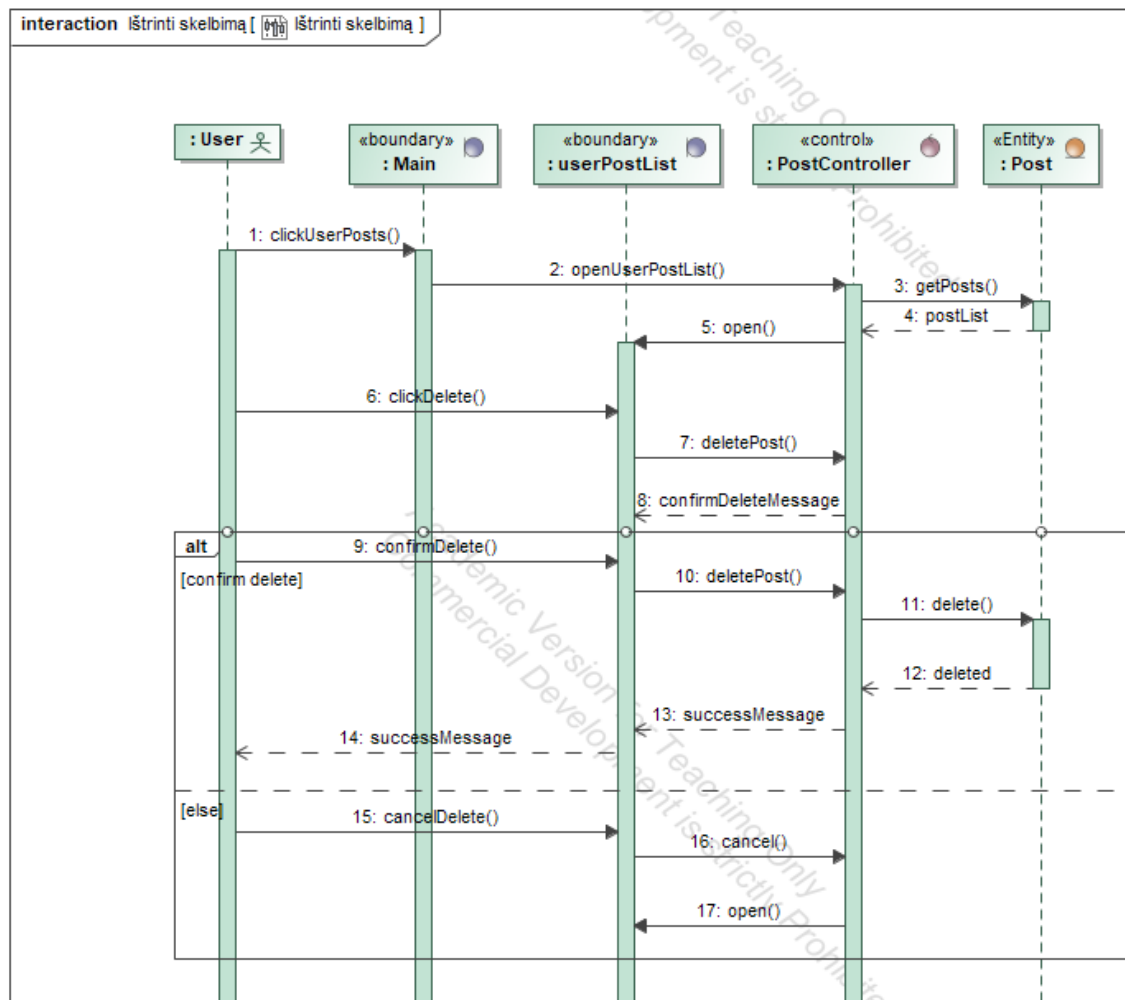
14 pav. Peržiūrėti įkeltų skelbimų sąrašą sekų diagrama



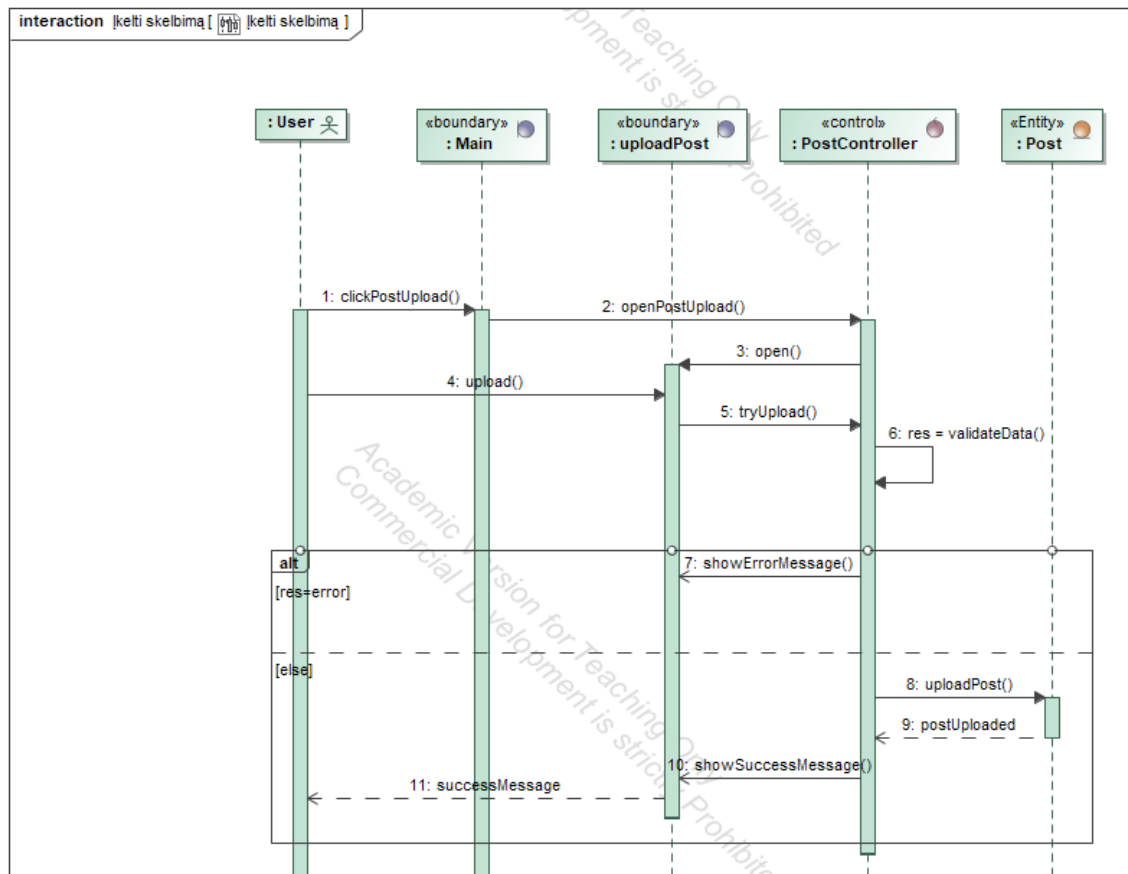
15 pav. Pranešti apie skelbimo suaktyvinimą sekų diagrama



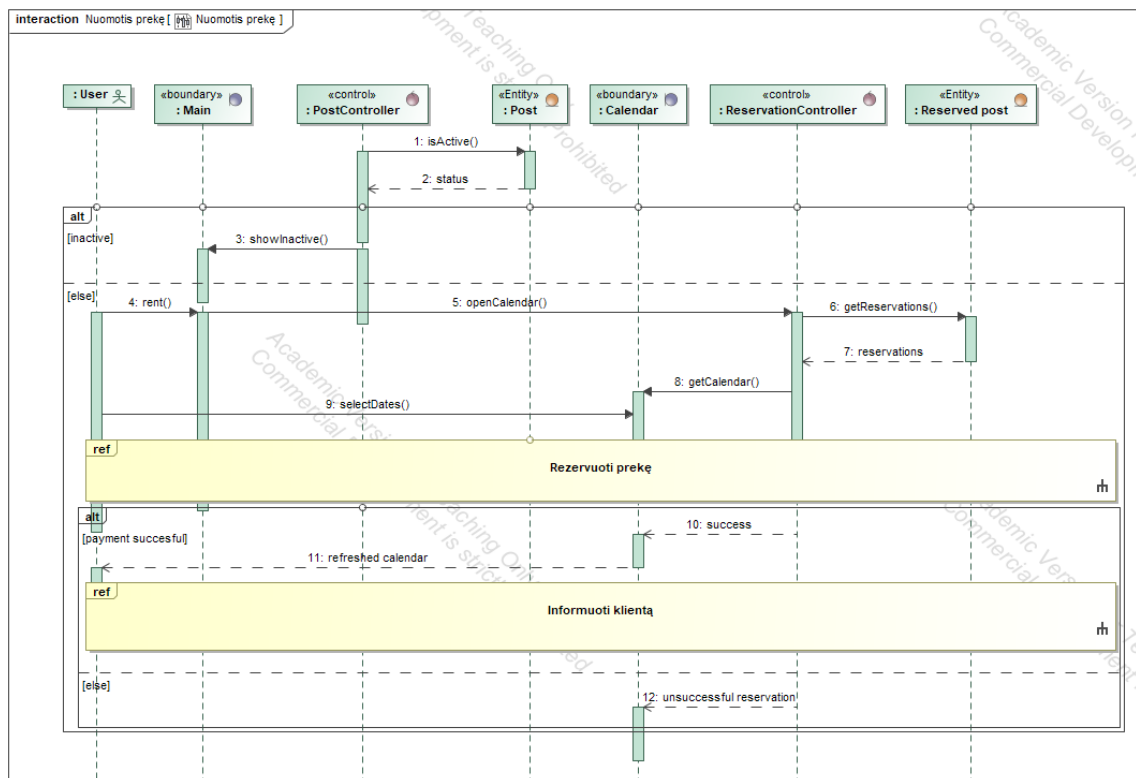
16 pav. Redaguoti skelbimų sekų diagrama



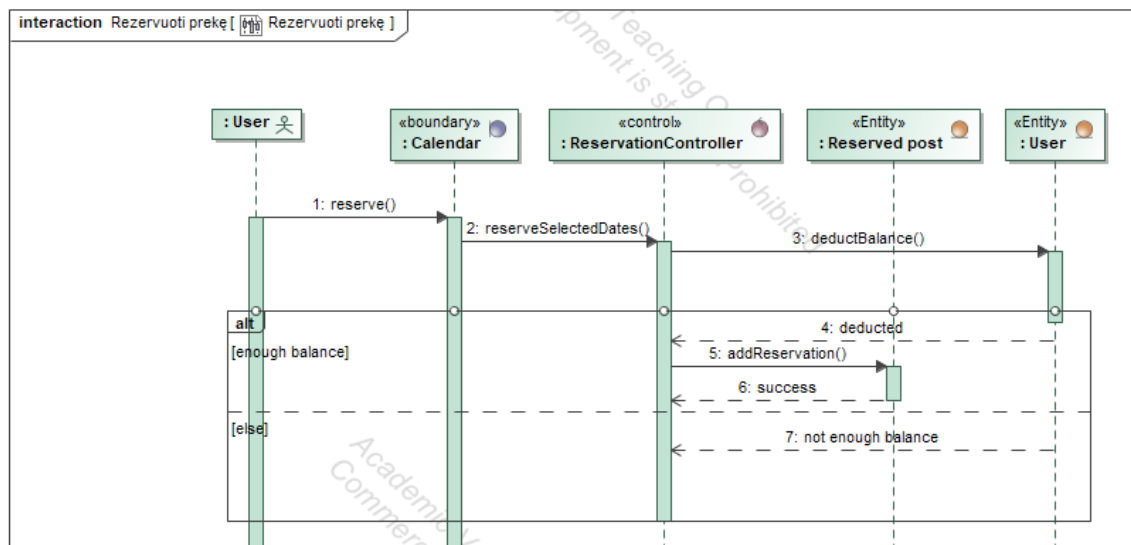
17 pav. Ištrinti skelbimą sekų diagrama



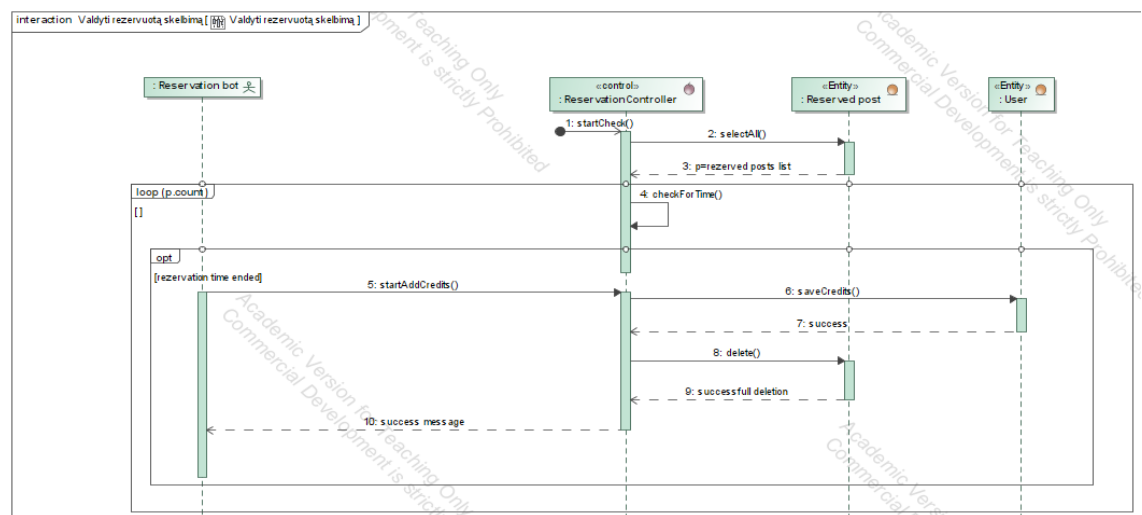
18 pav. Įkelti skelbimą sekų diagrama



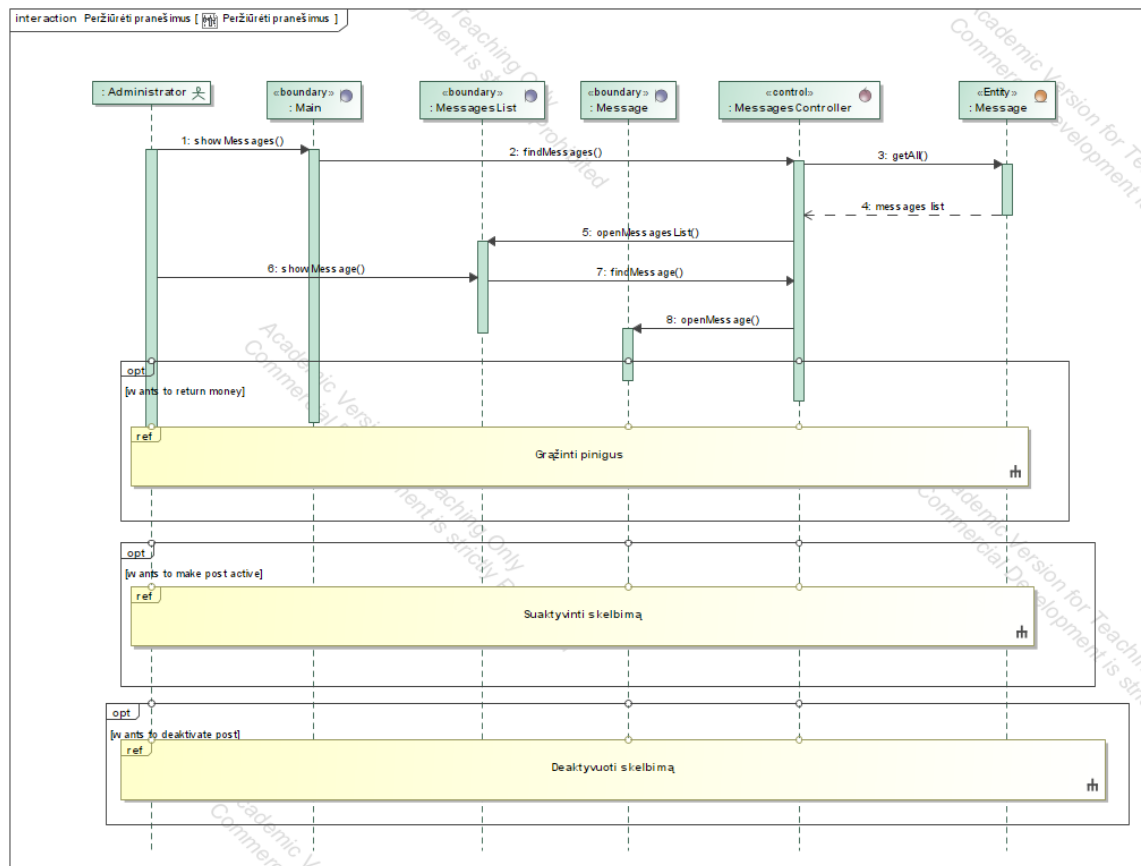
19 pav. Nuomoti prekę sekų diagrama



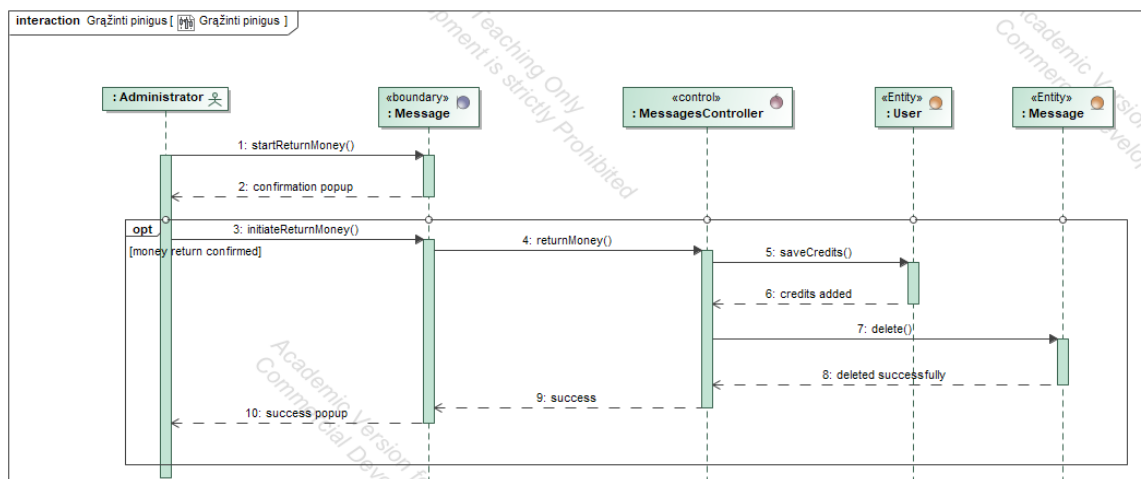
20 pav. Rezervuoti prekę sekų diagrama



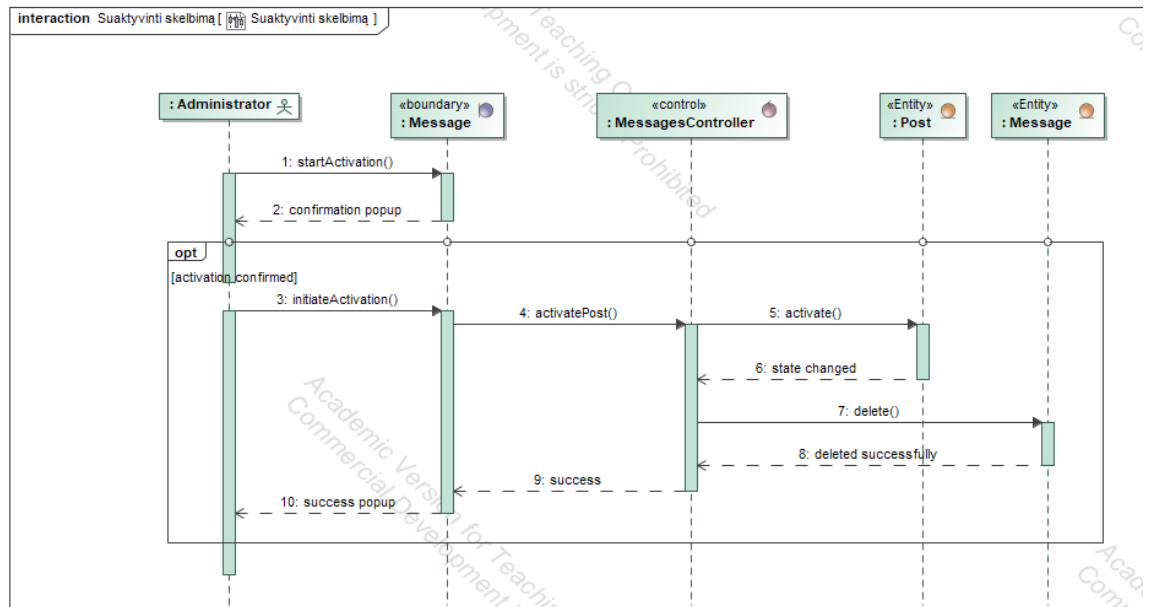
21 pav. Valdyti užrezervuotą skelbimą sekų diagrama



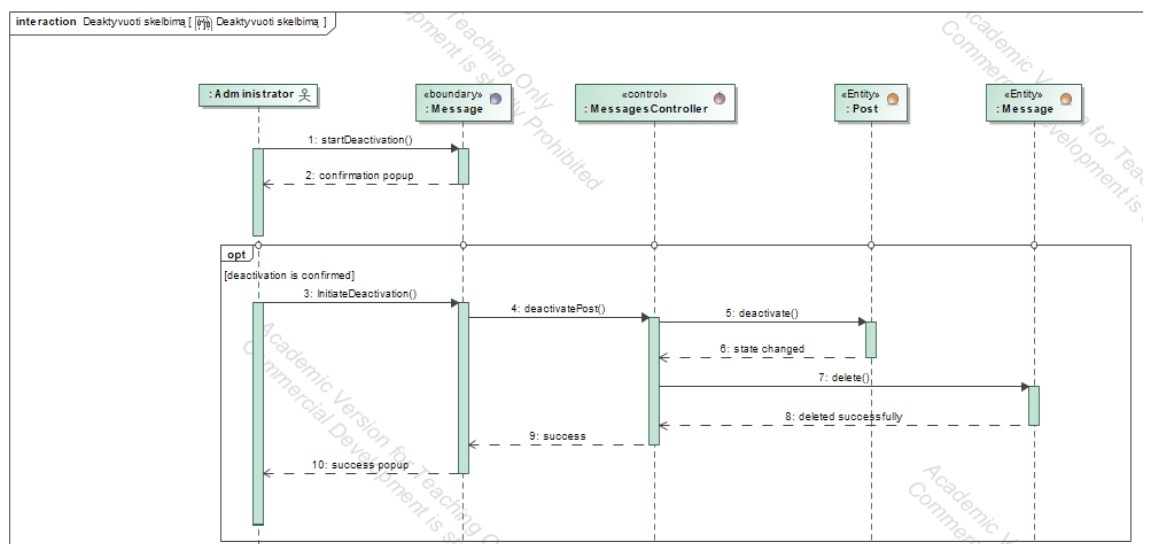
22 pav. Peržiūrėti pranešimus sekų diagrama



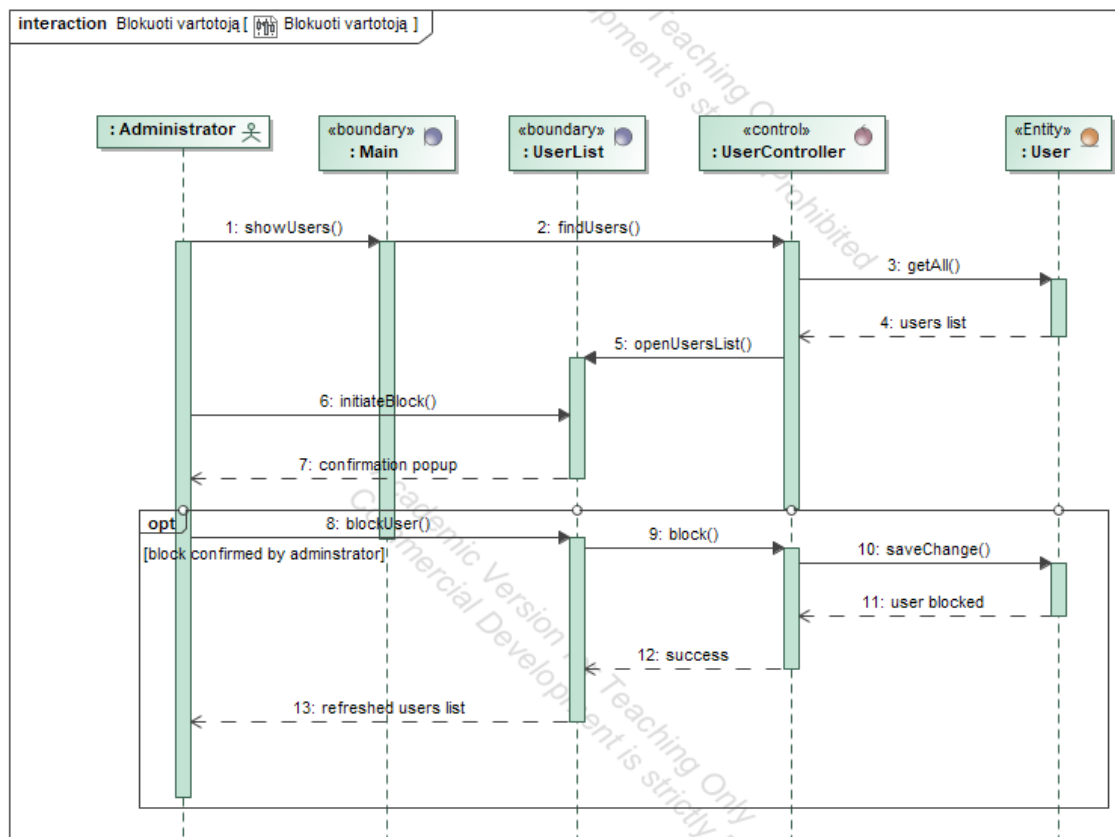
23 pav. Grąžinti pinigus sekų diagrama



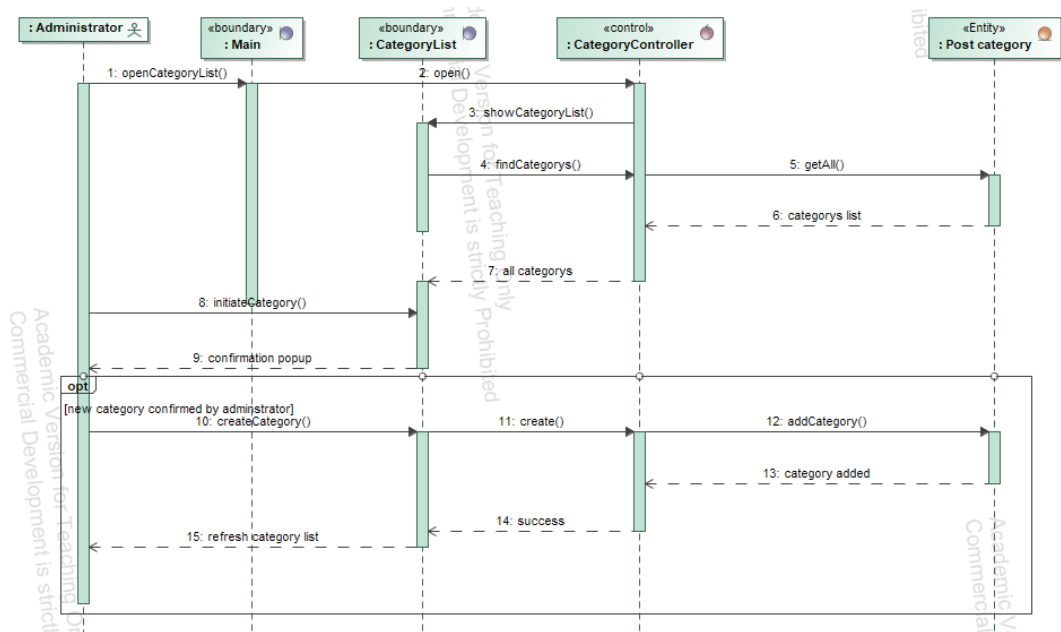
24 pav. Suaktyvinti skelbimą sekų diagrama



25 pav. Deaktyvuoti skelbimą sekų diagrama

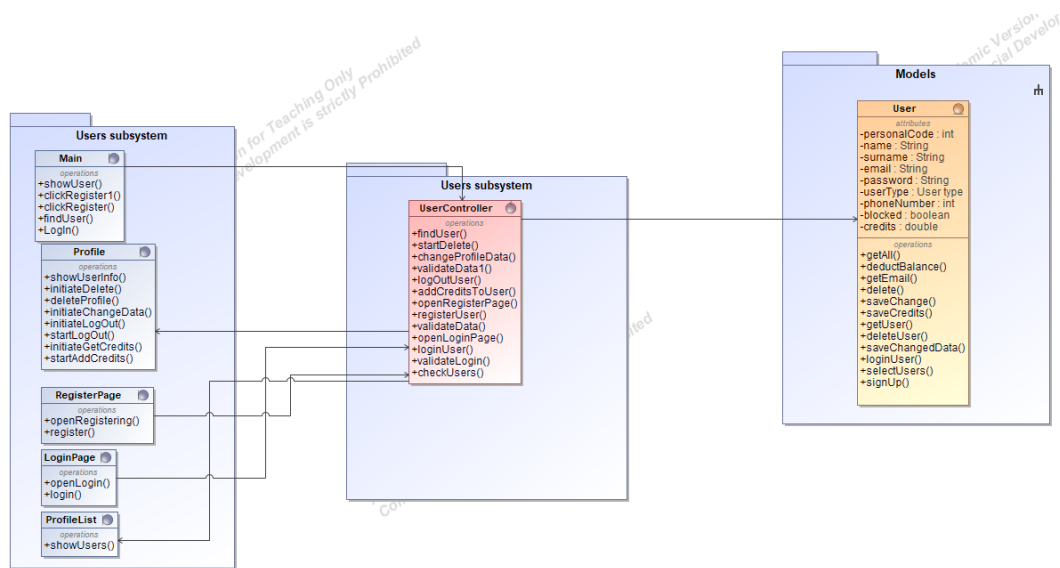


26 pav. Blokuoti vartotoją sekų diagrama

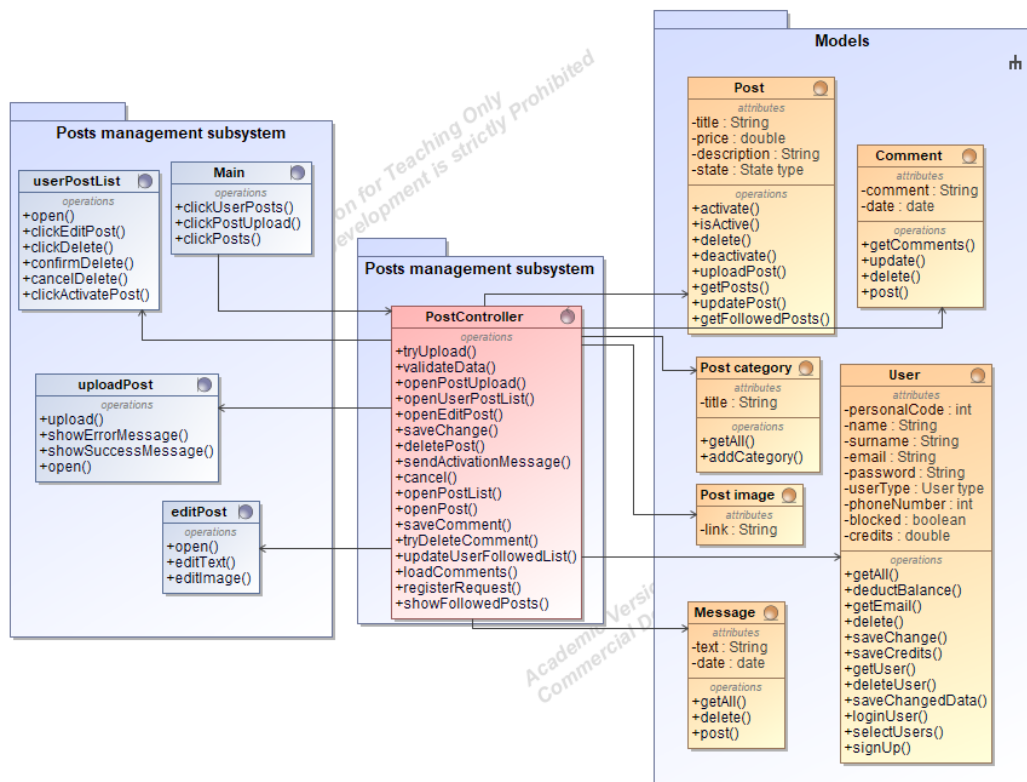


27 pav. Kurti skelbimų kategorijas sekų diagrama

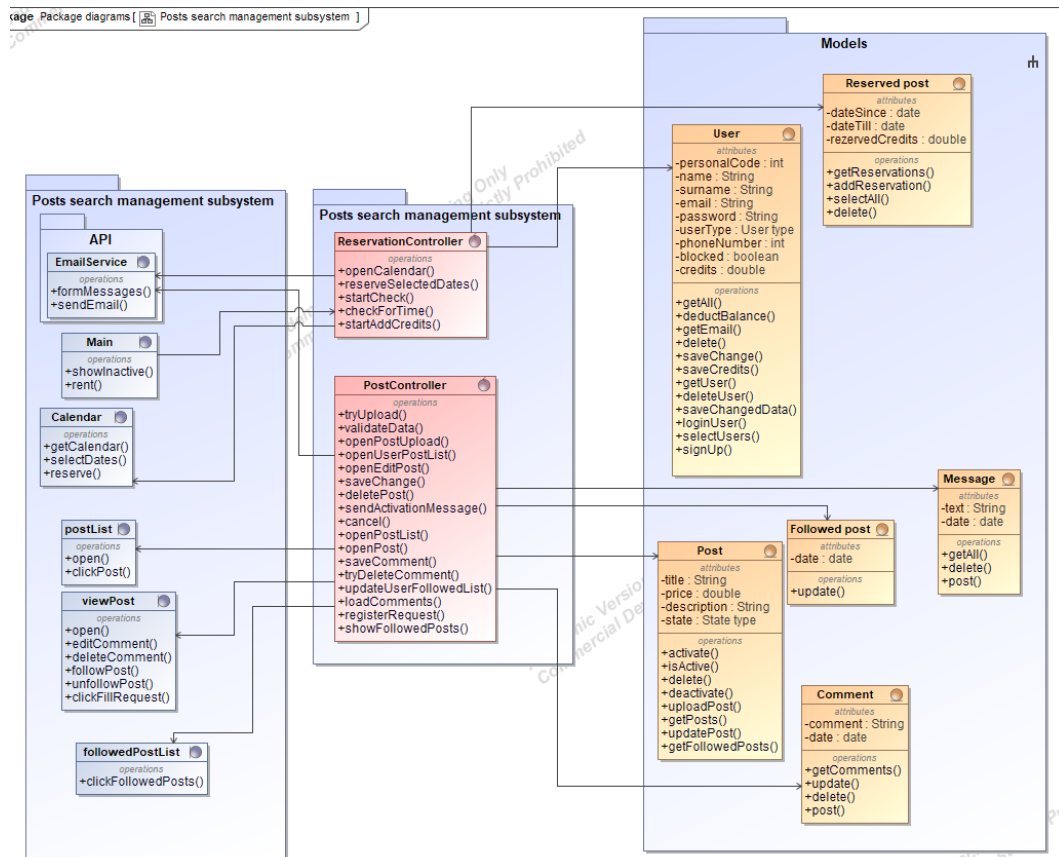
4.3 Sistemos klasių diagrama



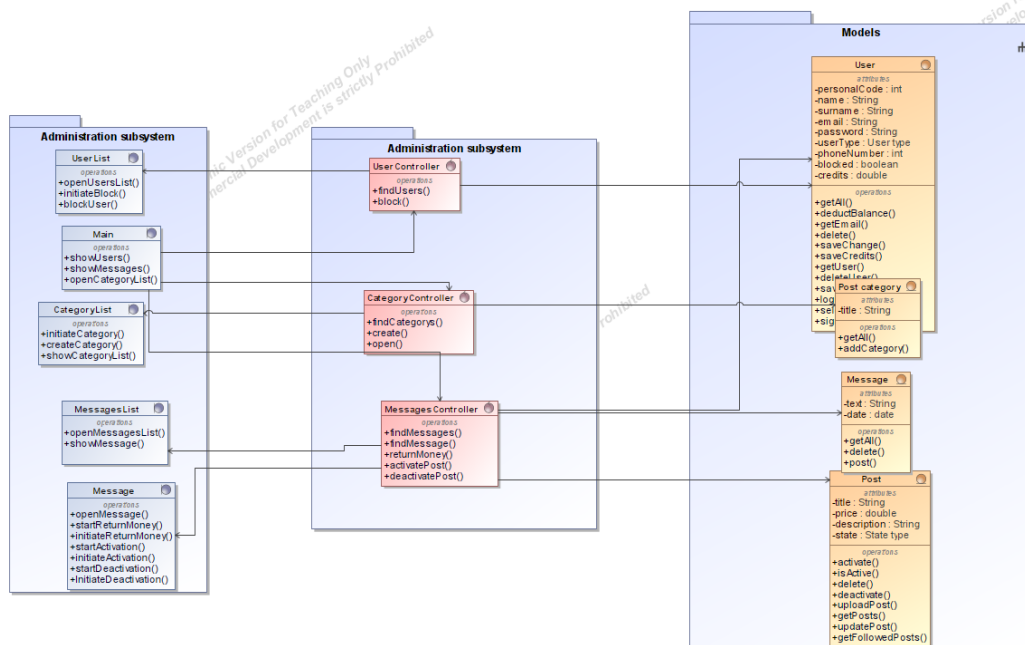
28 pav. Vartotojo posistemės klasių diagrama



29 pav. Skelbimų posistemės klasių diagrama

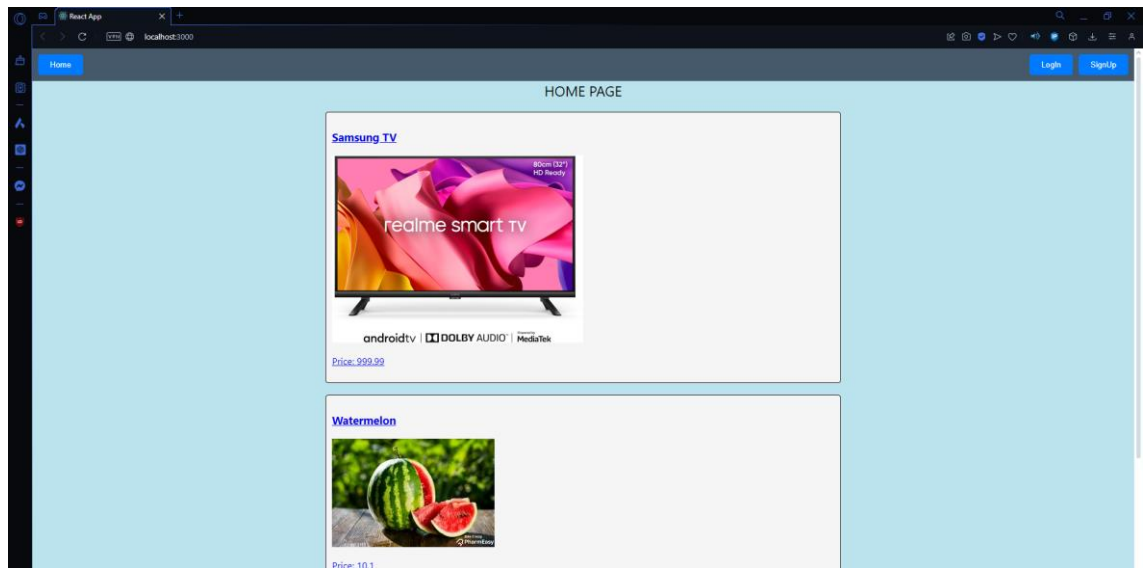


30 pav. Skelbimų paieškos posistemės klasių diagrama



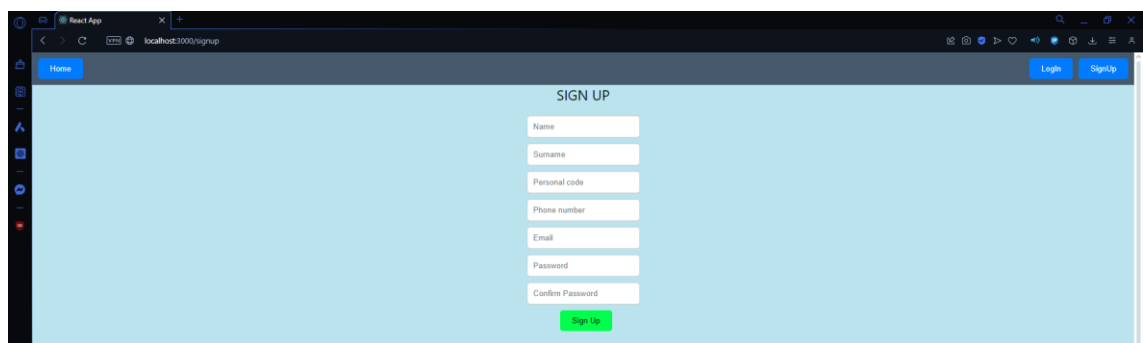
31 pav. Administracijos posistemės klasių diagrama

4.4 Sistemos prototipas



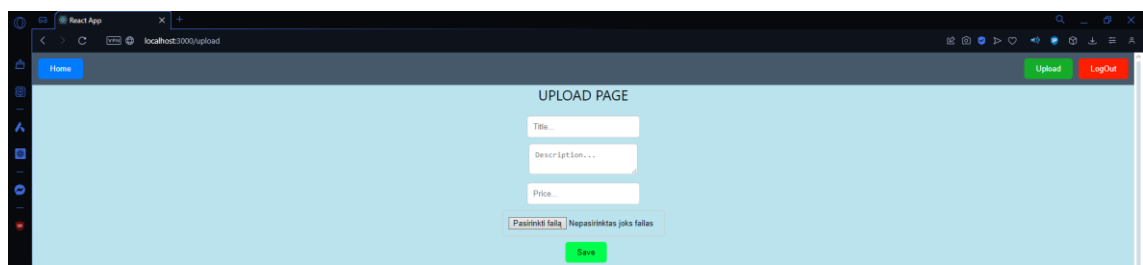
32 pav. Realizuotas pagrindinis puslapis

Tai yra pagrindinis puslapis, skirtas pateikti sąrašą skelbimų apsilankiusiems lankytojams, kuriame jie mato vartotojų patalpintus skelbimus bei galimybę užsiregistruoti arba prisijungti prie svetainės.



33 pav. Realizuotas registracijos puslapis

Tai yra registracijos puslapis, kuriame svečiai gali užregistruoti paskyrą. Pabaigus registracijos procesą naujo vartotojo duomenys yra išsaugomi, jam suteikiama galimybė prisijungti prie svetainės. Prisijungęs vartotojas gali į svetainę kelti savo skelbimus.



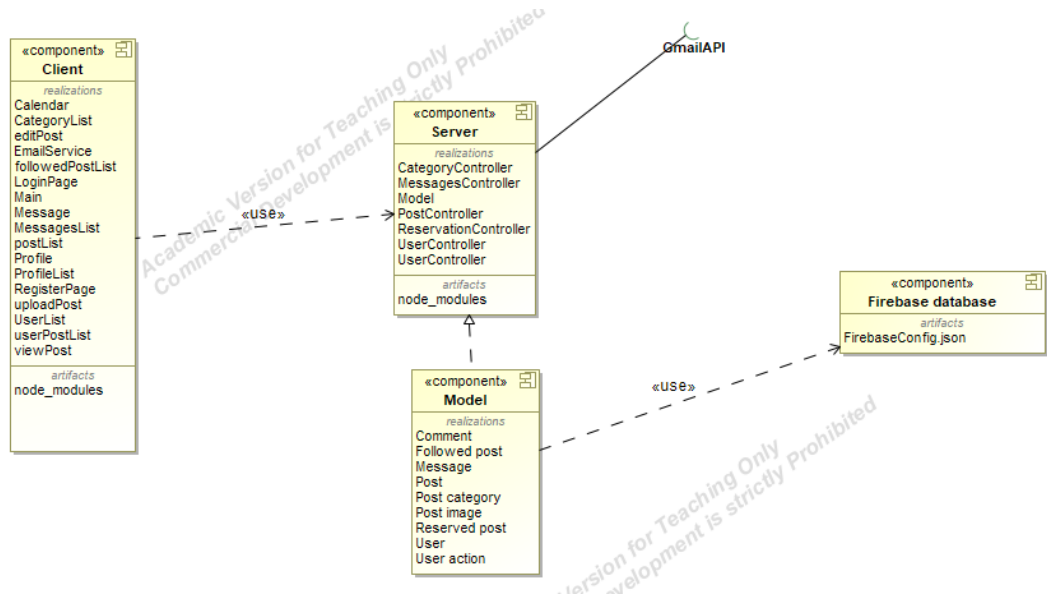
34 pav. Realizuotas skelbimo įkėlimo puslapis

Tai yra skelbimo įkėlimo puslapis. Prisijungęs vartotojas gali svetainėje patalpinti skelbimą, užpildęs visus duomenų laukus, skelbimo įkėlimo formoje.

Galime pastebėti, jog sukurtuose puslapiuose trūksta pagal reikalavimus apibrėžto funkcionalumo, bet tai bus papildyta ateityje. O lyginant sukurtus puslapius su pradiniais eskizais, kuriuos galima pamatyti 2 dokumento punkte, skirtumo didelio nėra.

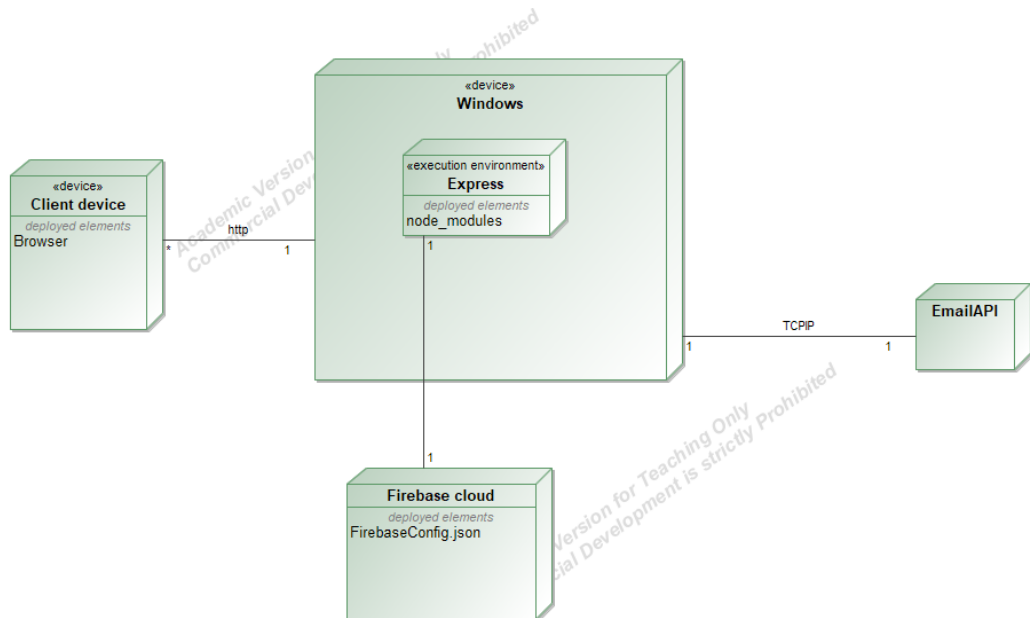
5 Sistemos realizacijos modelis

5.1 Komponentų diagrama



35 pav. Komponentų diagramos schema

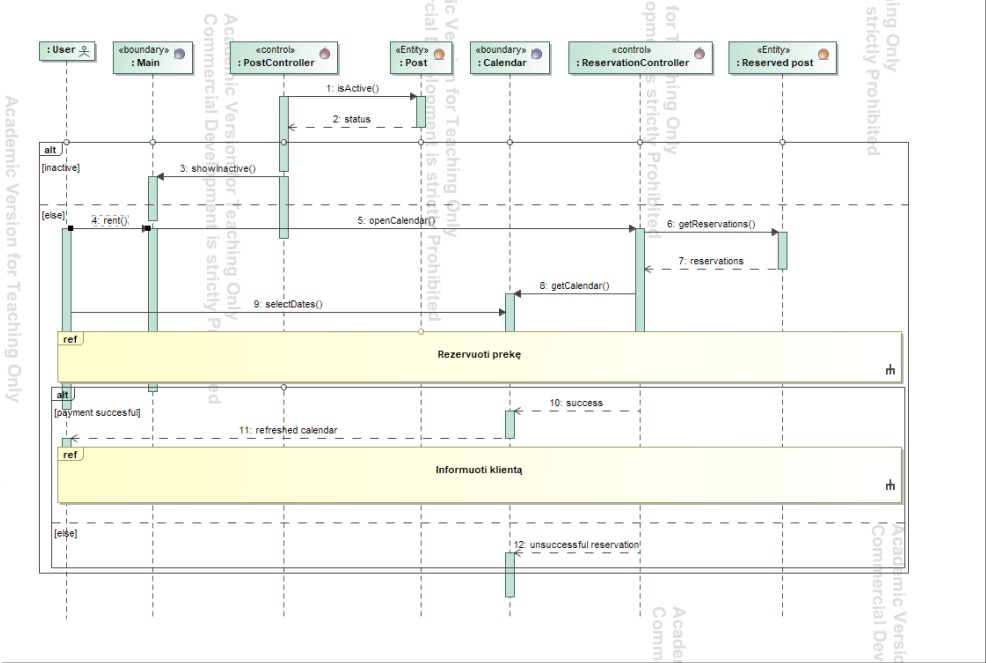
5.2 Diegimo diagrama



36 pav. Diegimo diagramos schema

6 Sistemos realizacija

6.1 Sekų diagramos kodo realizavimas



37 pav. Rezervavimo sekų diagrama

```
</div>  
<button className="reservation-button" onClick={openModal}>Rezervuoti</button>  
div>
```

38 pav. 4 žingsniui įvykdyti reikalingas mygtukas

```
const openCalendar = () => {  
  console.log("Open modal");  
  setShowModal(true);  
};
```

39 pav. 5 žingsniui įvykdyti reikalingas kalendoriaus atidarymas

```

<Modal
  isOpen={showModal}
  onRequestClose={closeModal}
  contentLabel="Select Reservation Date"
  className="modal"
  overlayClassName="modal-overlay"
>
  <h2>Pasirinkite datą</h2>
  {error && <p className="error-message">{error}</p>}
  <DatePicker
    selected={startDate}
    onChange={handleDateChange}
    startDate={startDate}
    endDate={endDate}
    selectsRange
    inline
    dayClassName={(date) => {
      const reserved = reservedDates.some((reservedDate) => {
        const reservedStart = new Date(reservedDate.startDate._seconds * 1000);
        const reservedEnd = new Date(reservedDate.endDate._seconds * 1000);
        return date >= reservedStart && date <= reservedEnd;
      });
      return reserved ? 'reserved-date' : null;
    }}
  />
  <button onClick={handleSelect}>Pasirinkti</button>
  <button onClick={closeModal}>Uždaryti</button>
</Modal>

```

40 pav. 5 žingsniui įvykdyti reikalingas kalendorius

```

useEffect(() => {
  const getReservations = async () => {
    try {
      const response = await axios.get(`http://localhost:5000/reservations/post/${postId}`);
      setReservedDates(response.data);
    } catch (error) {
      console.error("Error fetching reserved dates:", error);
    }
  };

  getReservations();
}, [postId]);

```

41 pav. 6-8 žingsniui įvykdyti reikalinga rezervacijų gavimas

```

async getReservationsByPostId(req, res) {
  try {
    const reservations = await Reservation.getReservationsByPostId(req.params.postId);
    res.status(200).json(reservations);
  } catch (error) {
    console.error("Error fetching reservations by post ID:", error);
    res.status(500).json({ success: false, error: error.message });
  }
}

```

42 pav. 6-8 žingsniui įvykdyti reikalinga rezervacijų gavimas

```
static async getReservationsByPostId(postId) {
  try {
    const snapshot = await firebaseAdmin.firestore().collection("reserved_post").where("fk_postID", '==', postId).get();
    const reservations = snapshot.docs.map(doc => ({ id: doc.id, ...doc.data() }));
    return reservations;
  } catch (error) {
    console.error("Error fetching reservations by post ID:", error);
    throw error;
  }
}
```

43 pav. 6-8 žingsniui įvykdyti reikalinga rezervacijų gavimas

```
const handleSelect = async () => {
  if (!startDate || !endDate) {
    setError("Pasirinkite pradžios ir pabaigos datas.");
    return;
  }

  const numberOfDays = Math.ceil((endDate - startDate) / (1000 * 60 * 60 * 24)) + 1;

  const totalPrice = parseFloat(post.price) * numberOfDays;

  const overlap = reservedDates.some((reservedDate) => {
    const reservedStart = new Date(reservedDate.startDate._seconds * 1000);
    const reservedEnd = new Date(reservedDate.endDate._seconds * 1000);
    return startDate <= reservedEnd && endDate >= reservedStart;
  });
};
```

44 pav. 9 žingsniui įvykdyti reikalinga datų pasirinkimas

```
const response = await axios.post('http://localhost:5000/reserve', {
  startDate,
  endDate,
  reservedCredits: totalPrice,
  fk_userID: userId,
  fk_postID: postId
});
```

45 pav. rezervuoti prekę 2

```
async subtractCredits(req, res) {
  try {
    const { userId } = req.params;
    const { amount } = req.body;

    if (typeof amount !== 'number' || amount <= 0) {
      return res.status(400).json({ success: false, message: "Invalid amount" });
    }

    const updatedCredits = await User.subtractCredits(userId, amount);
    res.status(200).json({ success: true, updatedCredits });
  } catch (error) {
    console.error("Error subtracting credits:", error);
    res.status(500).json({ success: false, error: error.message });
  }
}
```

46 pav. 3-4 žingsnyje reikalingas balanso atėmimas

```

static async subtractCredits(userId, amount) {
  try {
    const userRef = firebaseAdmin.firestore().collection("users").doc(userId);
    const docSnapshot = await userRef.get();

    if (!docSnapshot.exists) {
      throw new Error("User not found");
    }

    const userData = docSnapshot.data();
    const currentCredits = userData.credits || 0;
    const newCredits = currentCredits - amount;

    await userRef.update({ credits: newCredits });

    return newCredits;
  } catch (error) {
    console.error("Error updating credits:", error);
    throw error;
  }
}

```

47 pav. 3-4 žingsnyje reikalingas balanso atėmimas

```

async addReservation(req, res) {
  try {
    const { startDate, endDate, reservedCredits, fk_userID, fk_postID } = req.body;
    const reservationId = await Reservation.addReservation({
      startDate,
      endDate,
      reservedCredits,
      fk_userID,
      fk_postID
    });
    res.status(200).json({ success: true, reservationId });
  } catch (error) {
    console.error("Error adding reservation:", error);
    res.status(500).json({ success: false, error: error.message });
  }
}

```

48 pav. 5-6 žingsnyje reikalingas balanso pridėjimas

```

static async addReservation(reservationData) {
  try {
    const { startDate, endDate, reservedCredits, fk_userID, fk_postID } = reservationData;
    const reservationId = firebaseAdmin.firestore().collection("reserved_post").doc().id;

    await firebaseAdmin.firestore().collection("reserved_post").doc(reservationId).set({
      startDate: firebaseAdmin.firestore.Timestamp.fromDate(new Date(startDate)),
      endDate: firebaseAdmin.firestore.Timestamp.fromDate(new Date(endDate)),
      reservedCredits,
      fk_userID,
      fk_postID
    });

    return reservationId;
  } catch (error) {
    console.error("Error adding reservation:", error);
    throw error;
  }
}

```

49 pav. 5-6 žingsnyje reikalingas balanso pridėjimas