

# „Fruit Bee Inc.“ Išmanusis šaldytuvas

Darbą atliko:

Vilius Minkevičius (darbo vadovas),  
Edvinas Šmita, Teodoras Šaulys,  
Tomas Mikna, Manfredas Šiurkus

# Turinys

- Sistemos užduotys
- Sistemos struktūra
- Kritinės projekto peržiūros rezultatai
- Testavimo planas ir scenarijai
- Techninė architektūra
- Sistemos realizacija:
  - Duomenų bazės schema
  - Įgyvendintas užduoties scenarijus

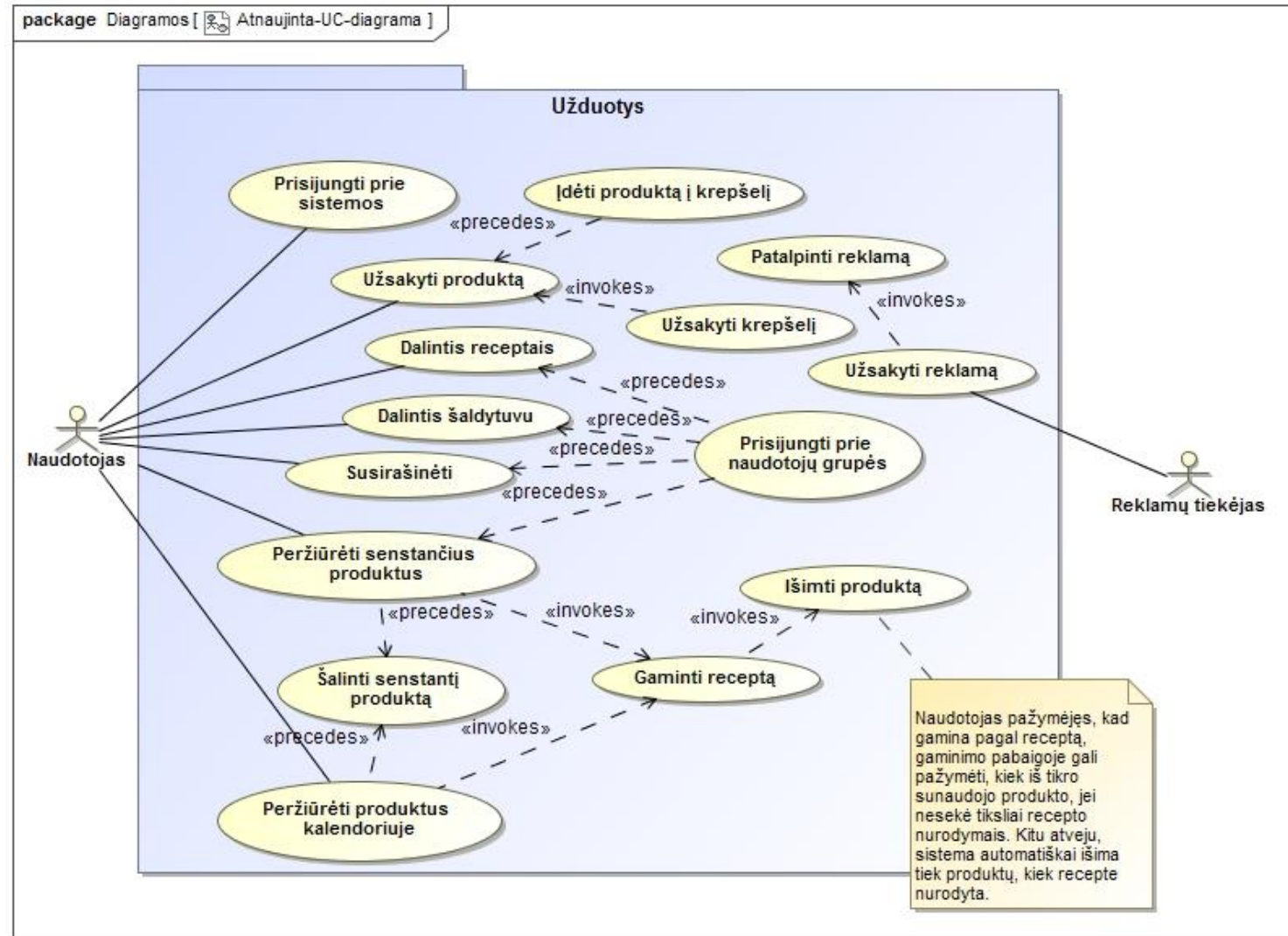
# Sistemos užduotys

- Užduočių sąrašas:
- Projektuojamos užduotys:
  - Pranešimo dėl produktų galiojimo pabaigos peržiūrėjimas
  - Produkto užsakymas per parduotuvę (nebuvo detalai projektuojama)
  - Šaldytuvo dalinimasis
  - Naudotojas bendrauja su kitais grupės nariais per pranešimus
  - Naudotojas atsiverčia kalendorių produktams peržiūrėti
  - Reklamų rodymas (įgyvendinama)

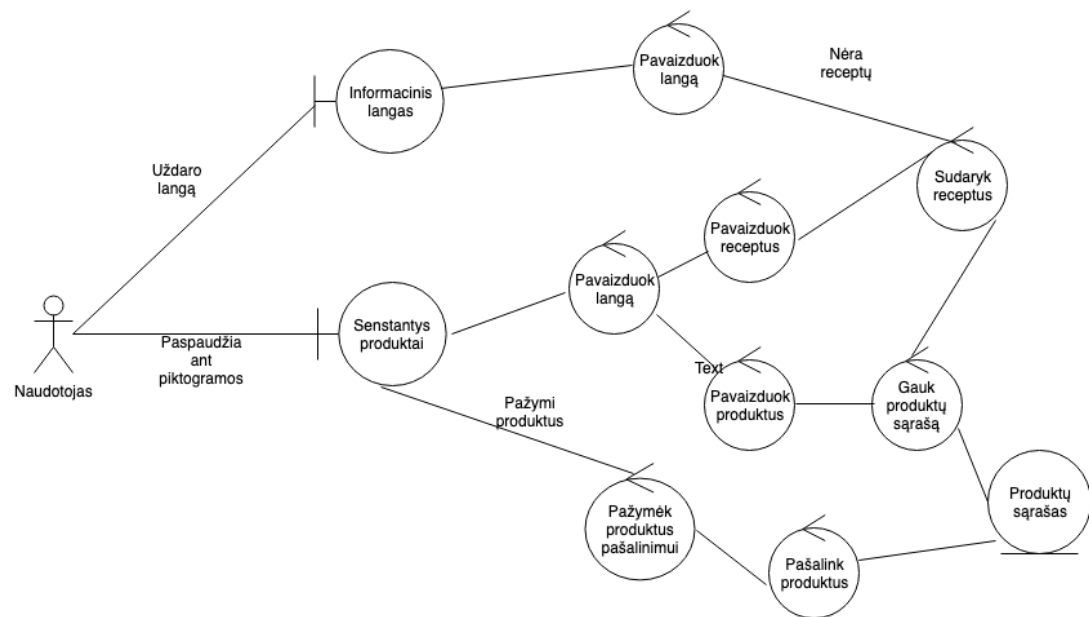
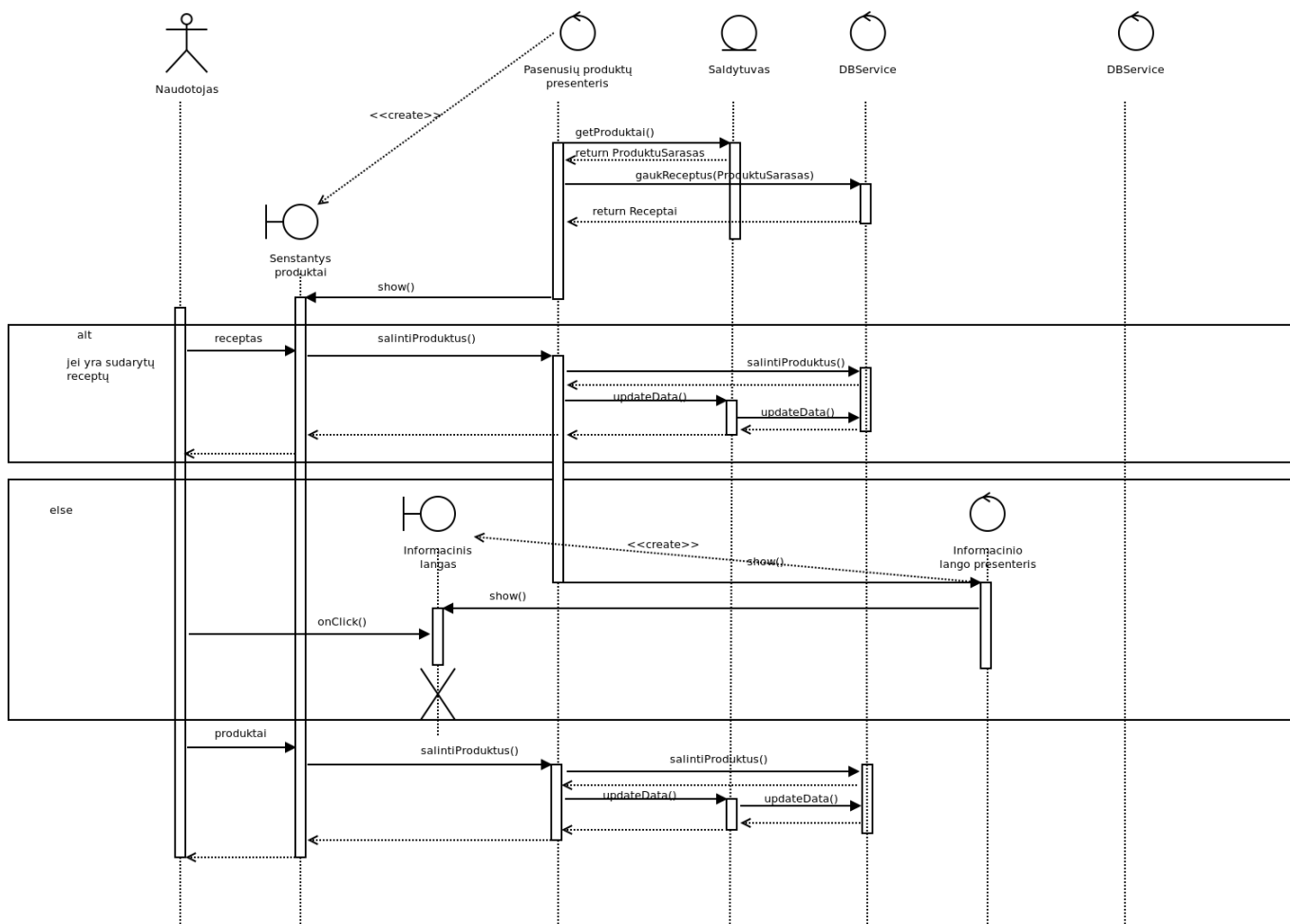
# Sistemos užduotys

- Numatomos, bet ne projektuojamos užduotys:
  - Reklamos užsakymas
  - Patalpinti reklamą
  - Prisijungti prie naudotojų grupės
  - Gaminti receptą
  - Šalinti sunaudotus produktus
  - Užsakyti krepšelį
  - Prisijungti prie sistemos

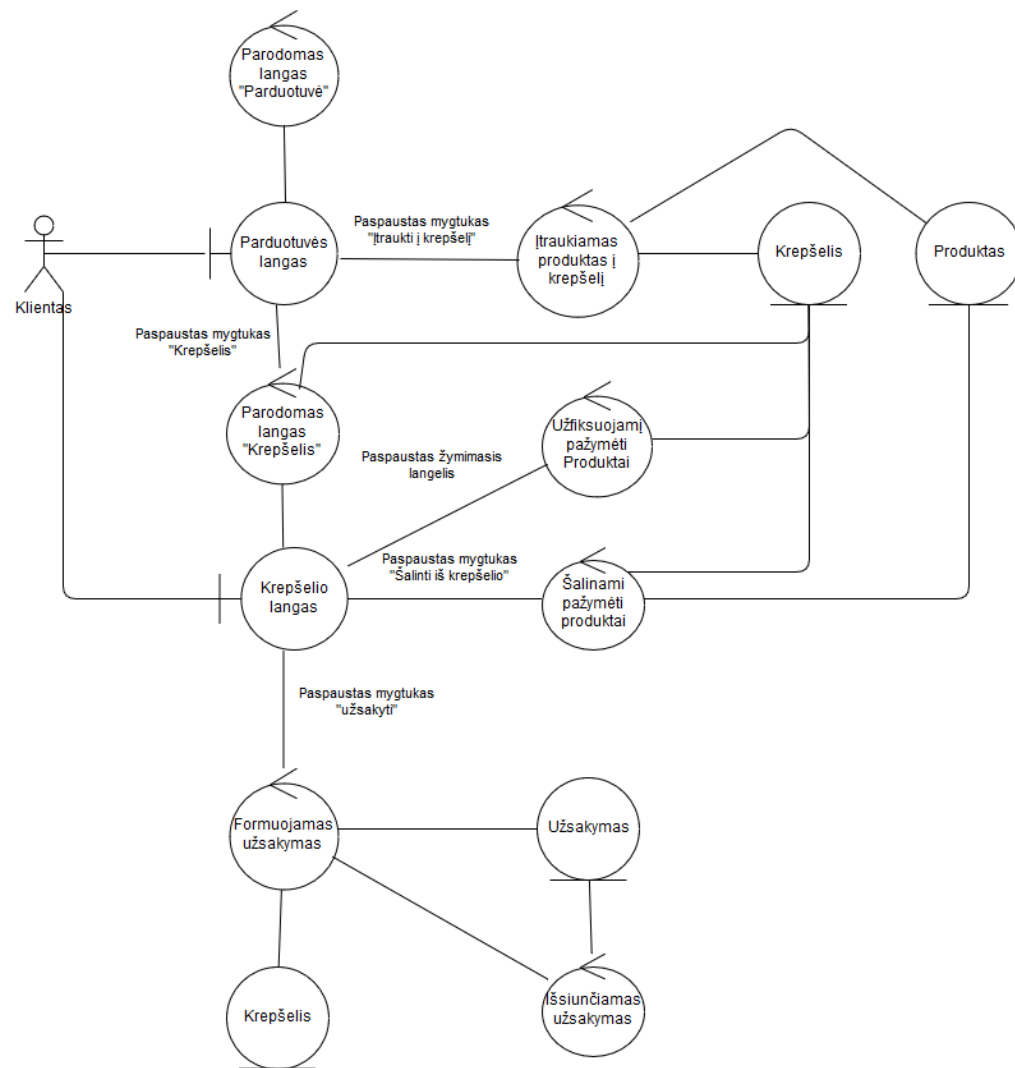
# Užduočių diagrama



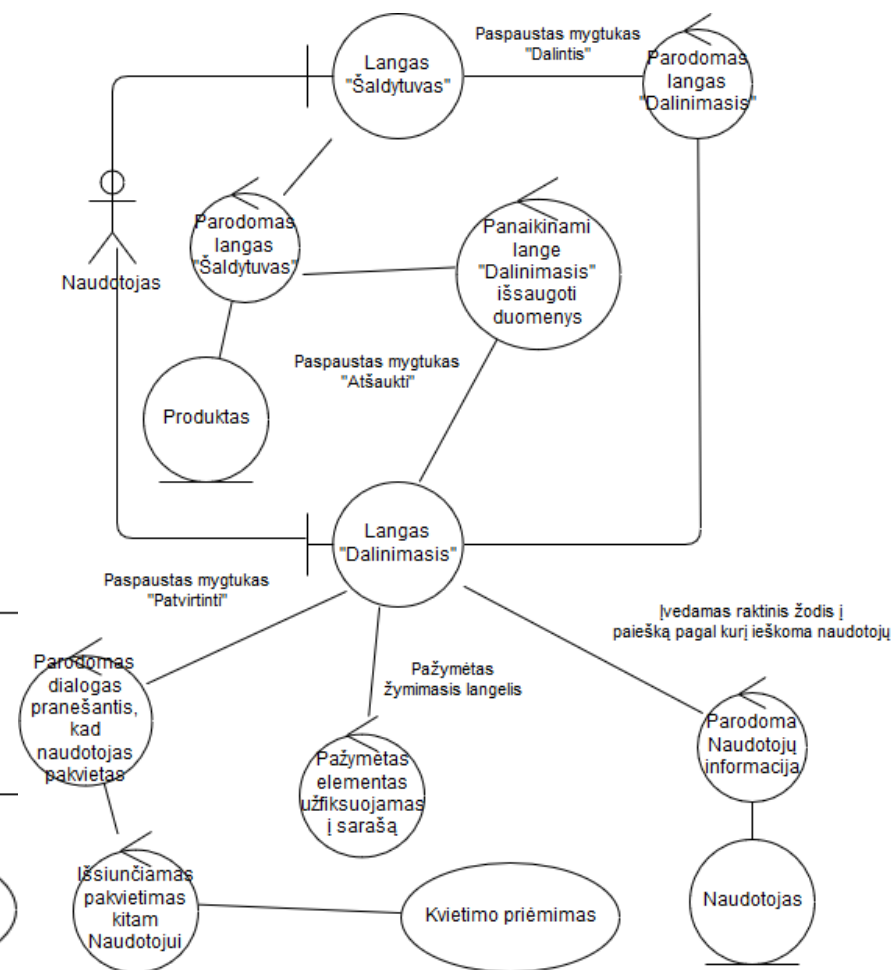
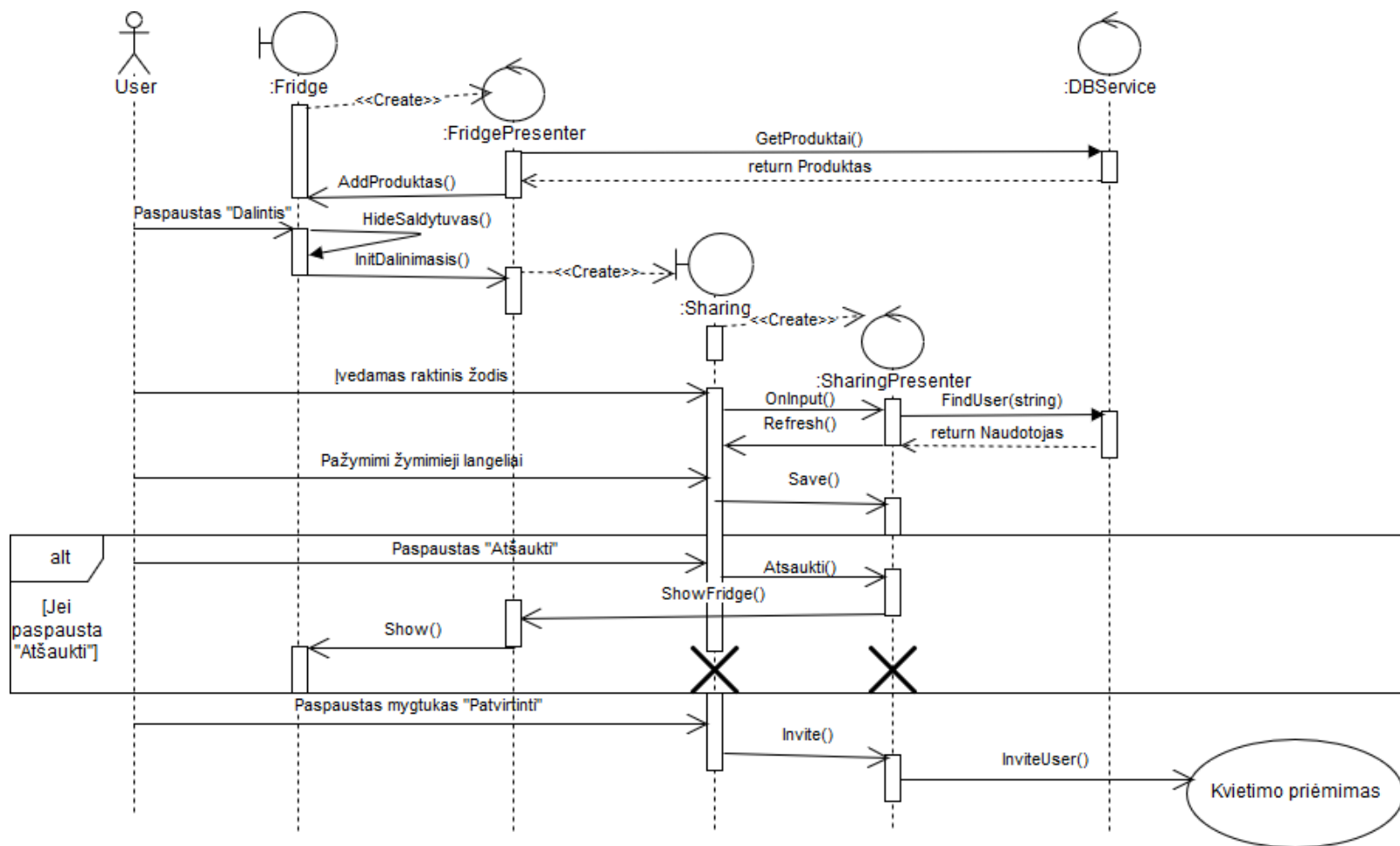
# Senstančių produktų pranešimo peržiūrėjimas



# Produkto užsakymas

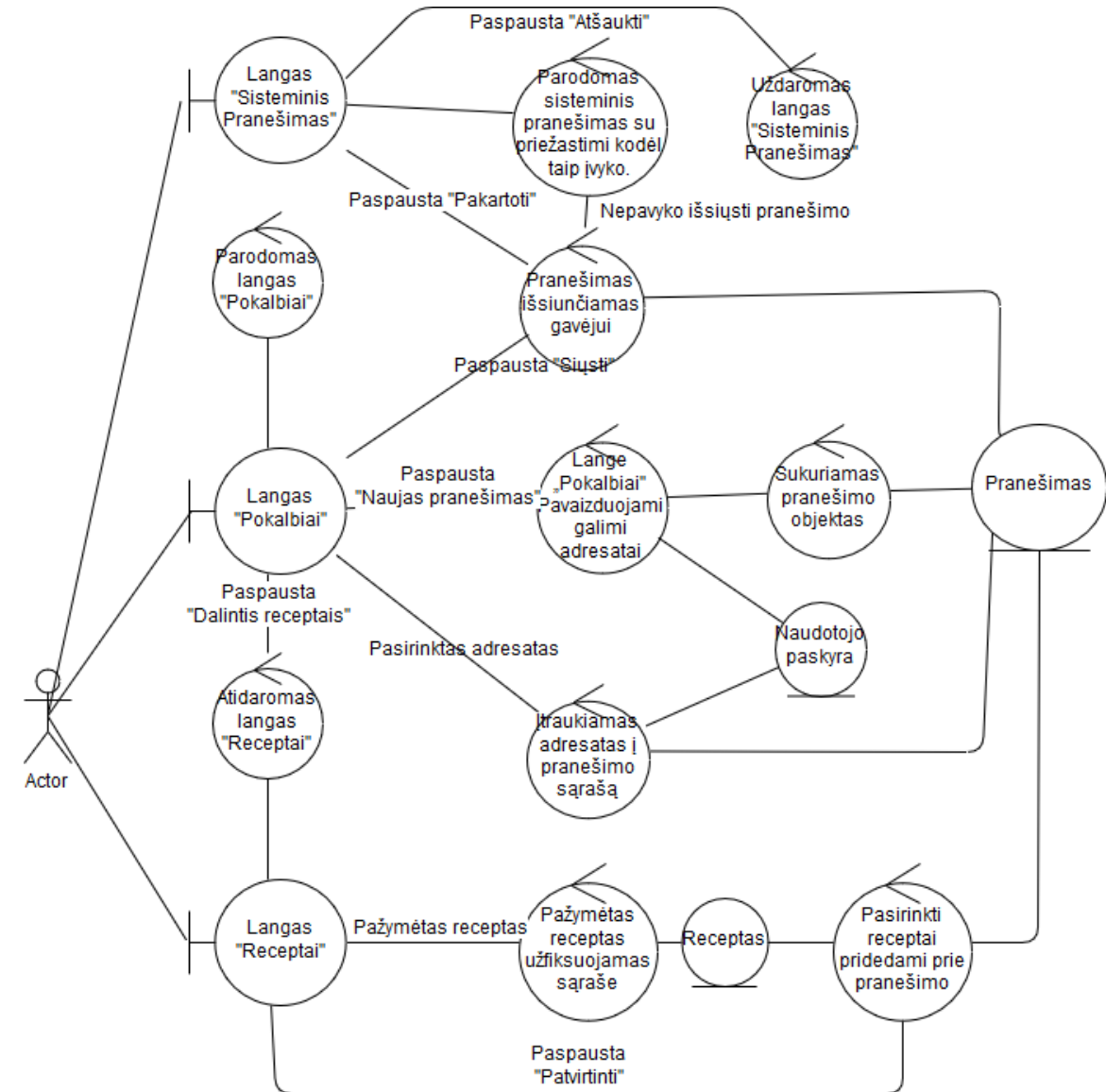
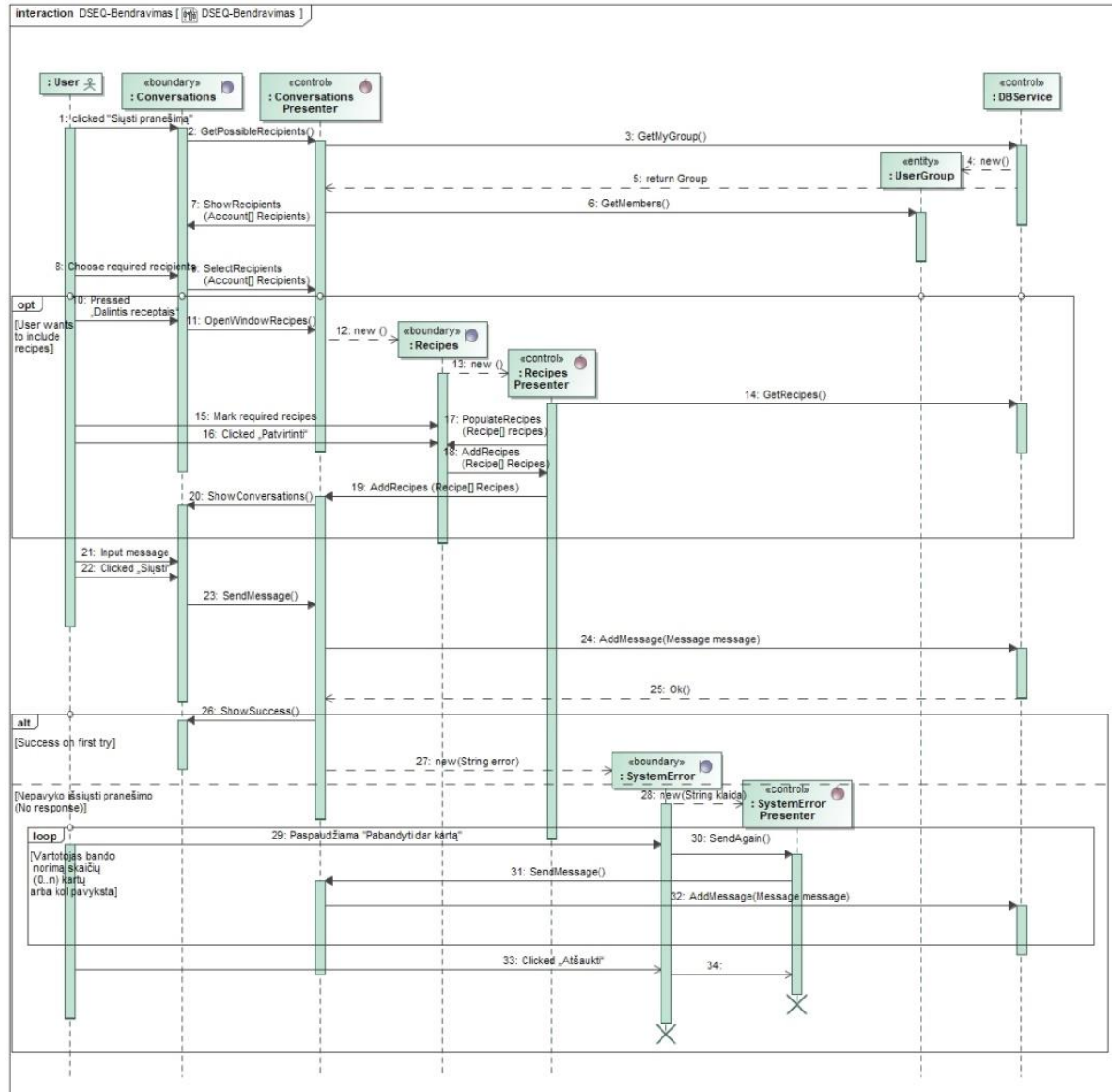


# Dalinimasis šaldytuvu

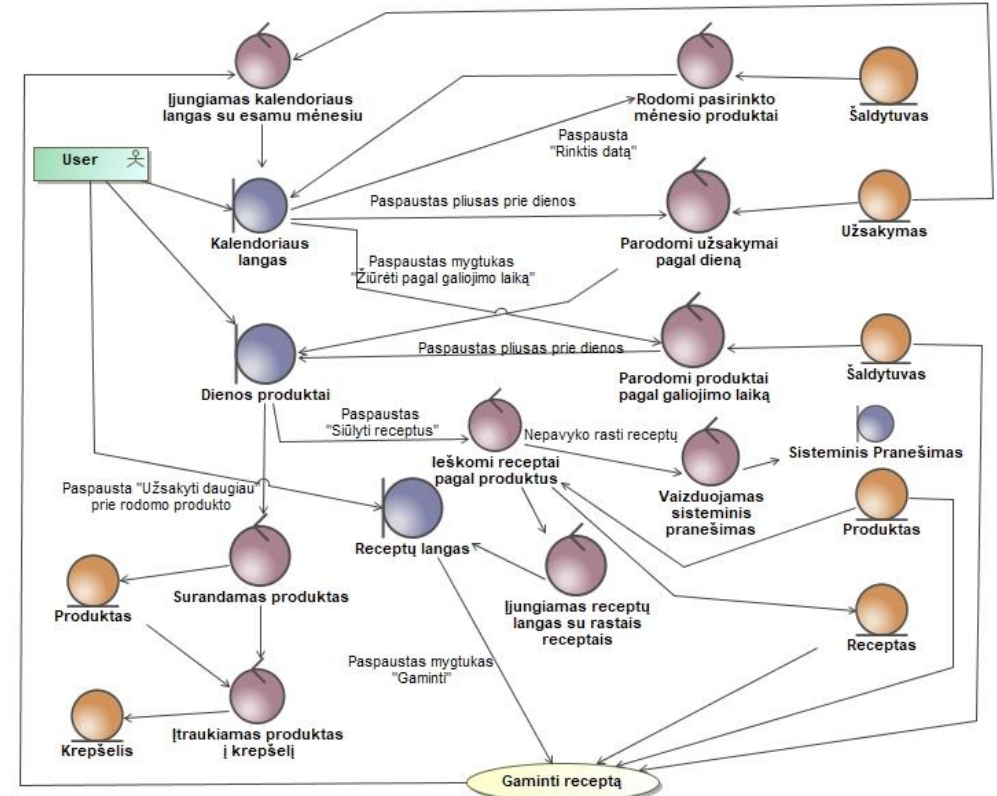
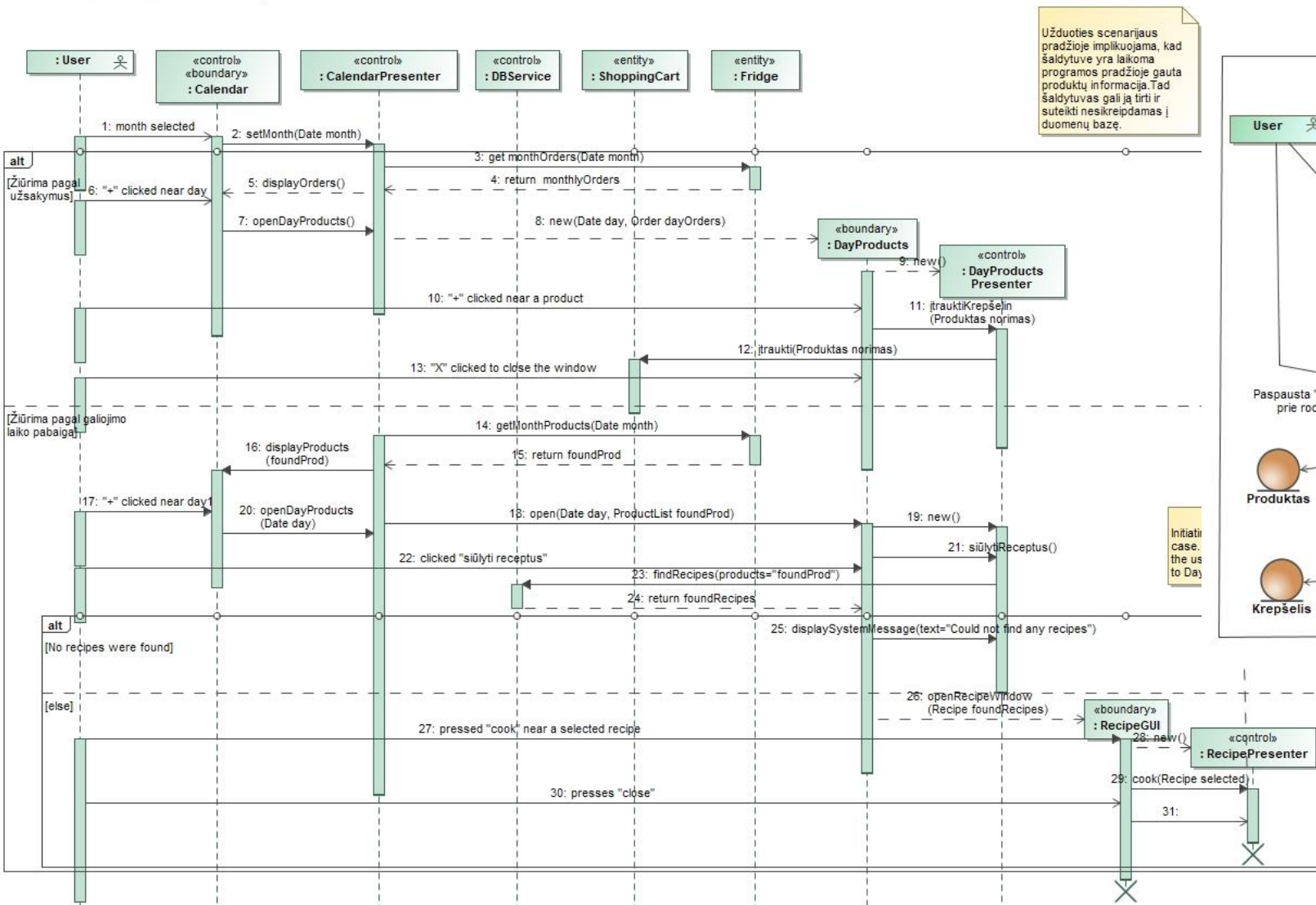




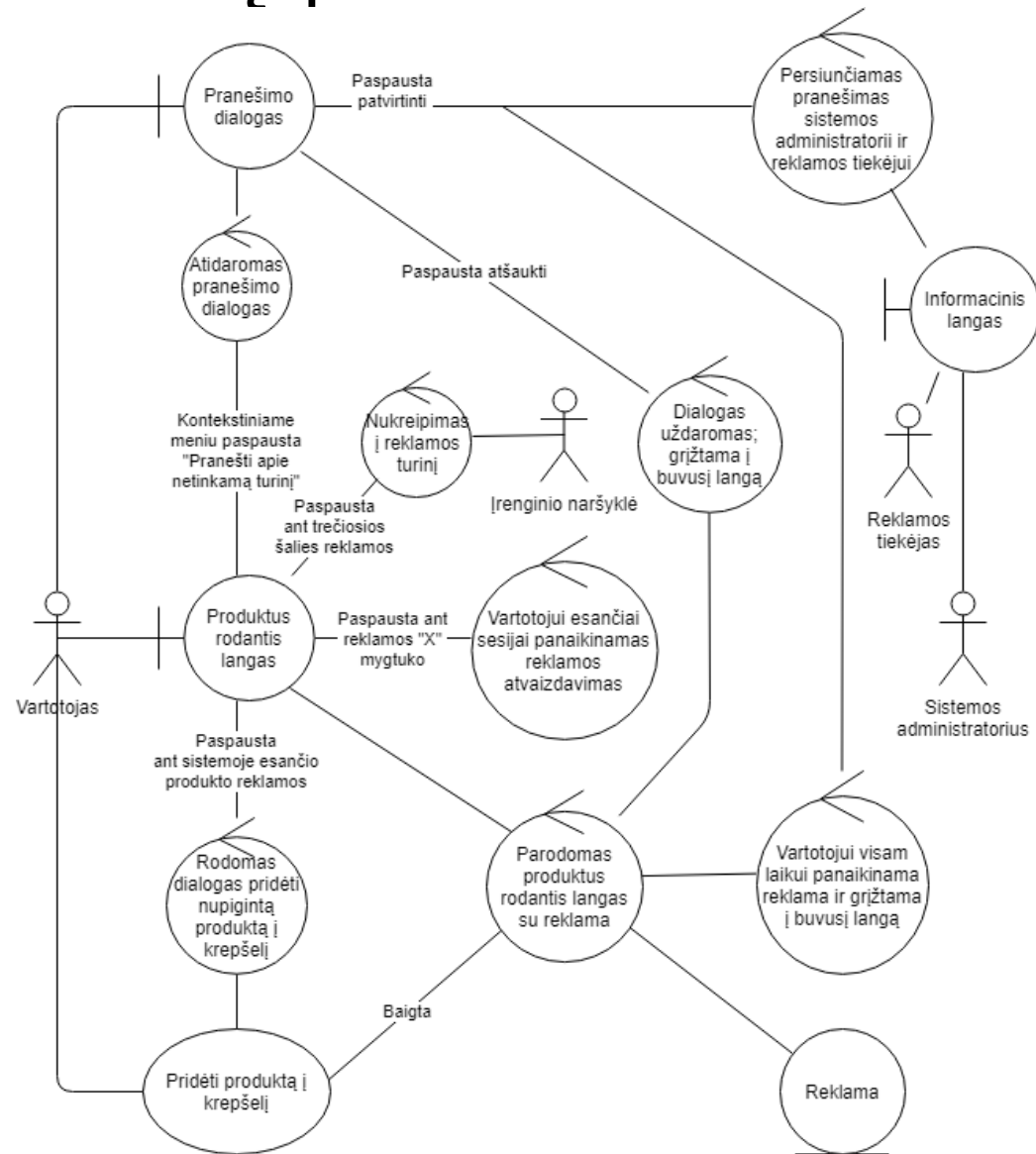
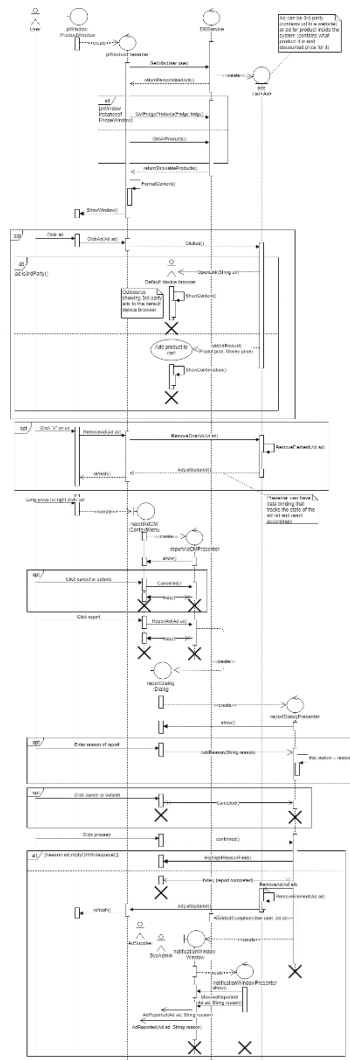
# Bendravimas pranešimais



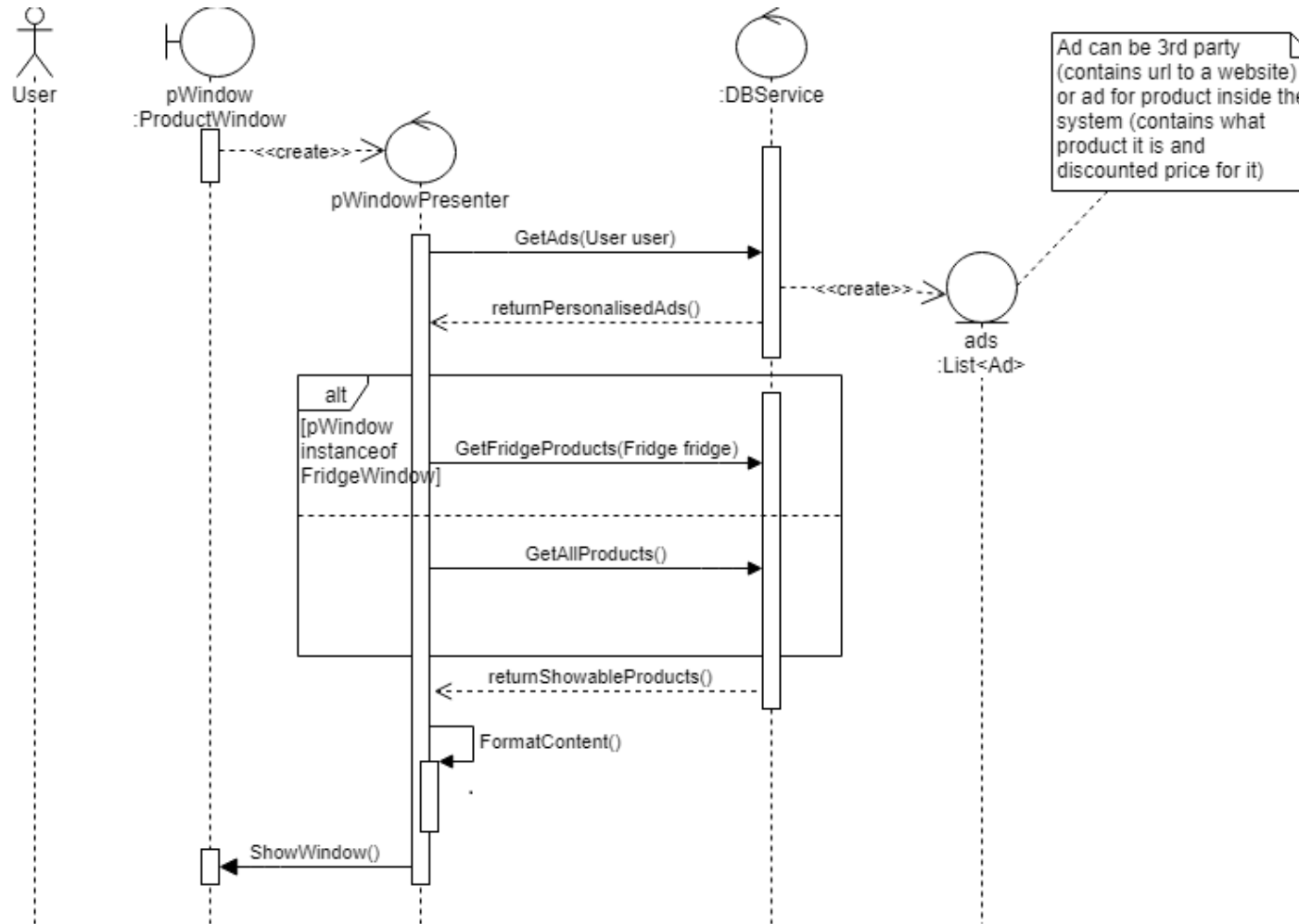
# Kalendoriaus naudojimas produktų peržiūrai

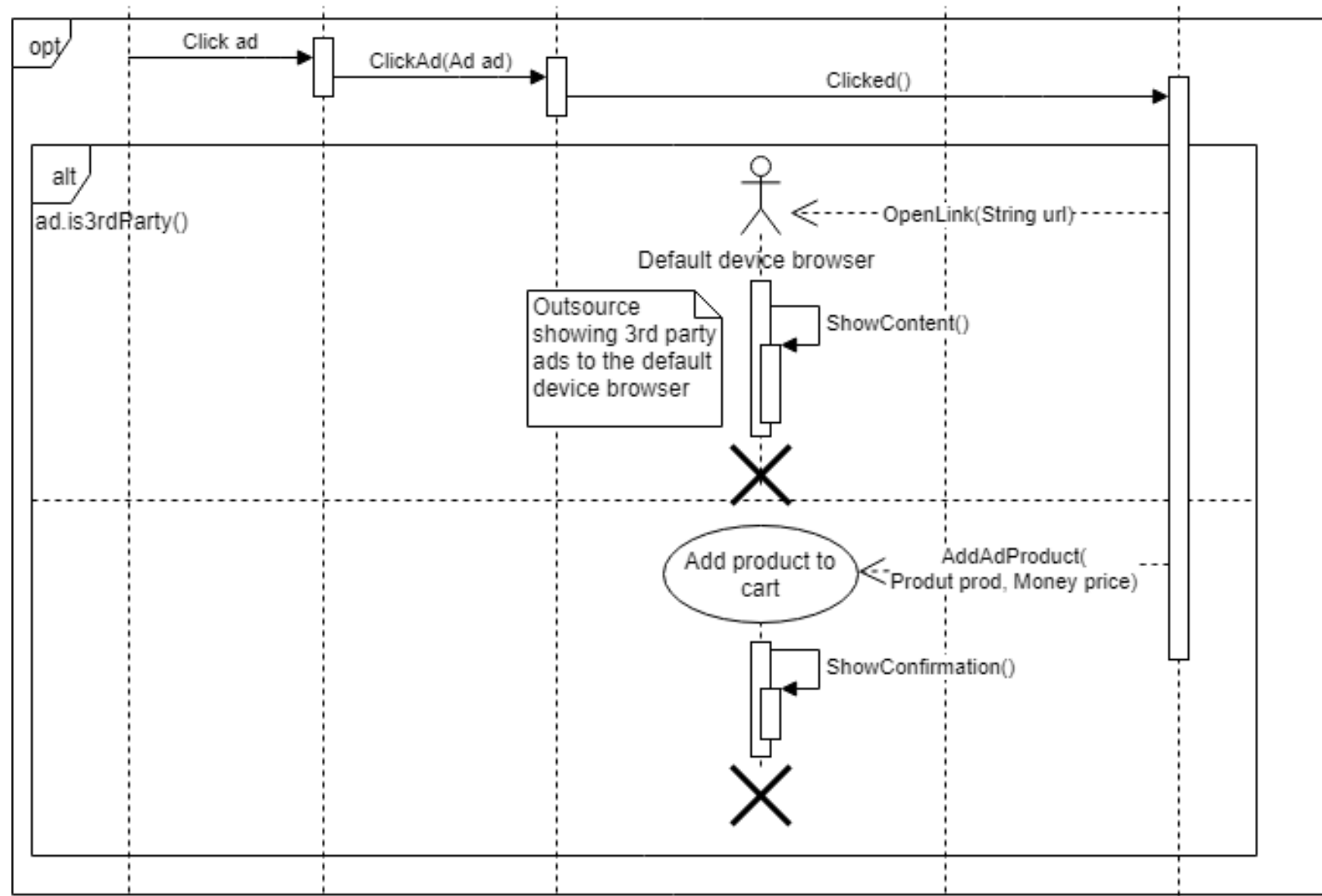


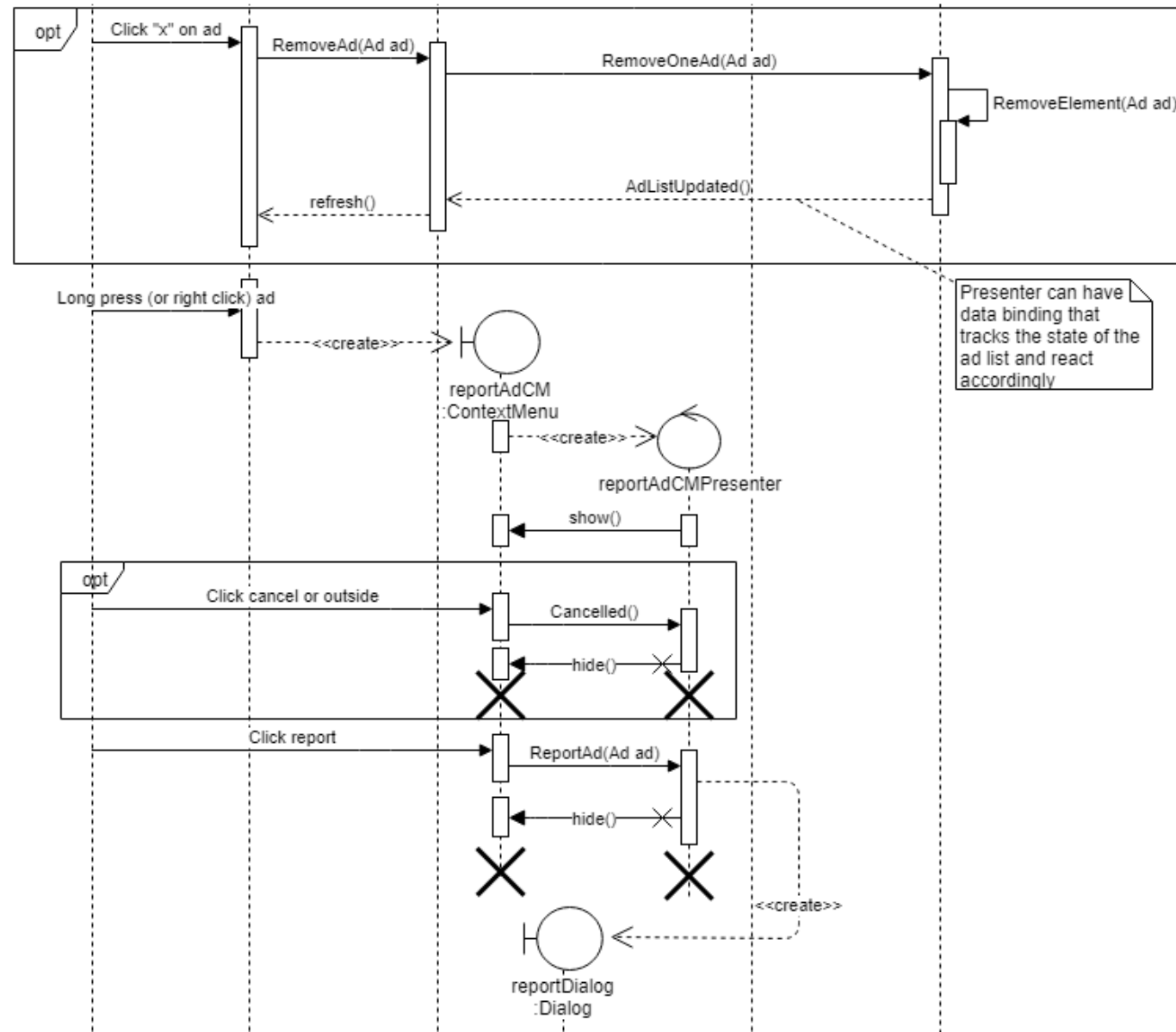
# Reklamų peržiūra

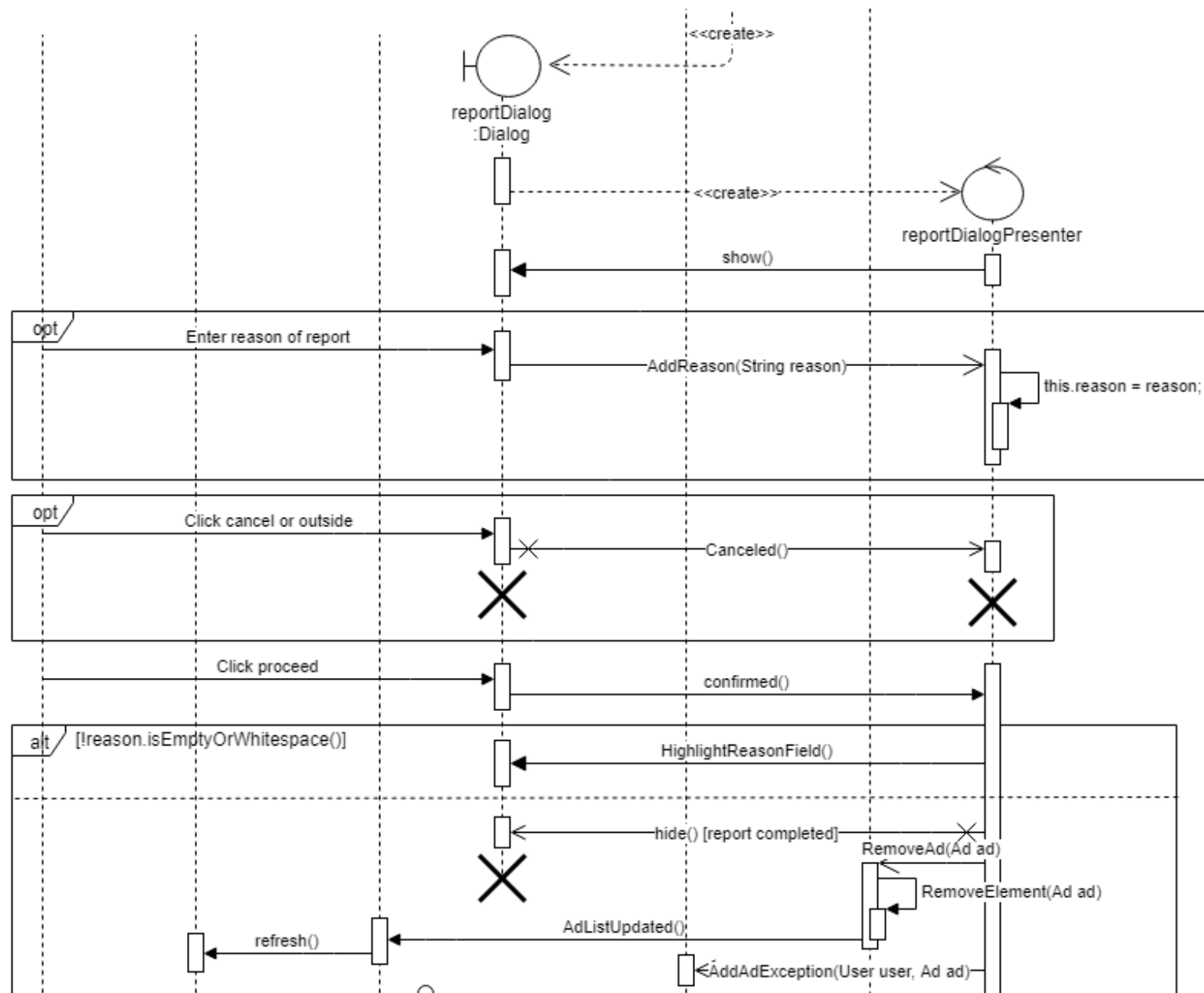


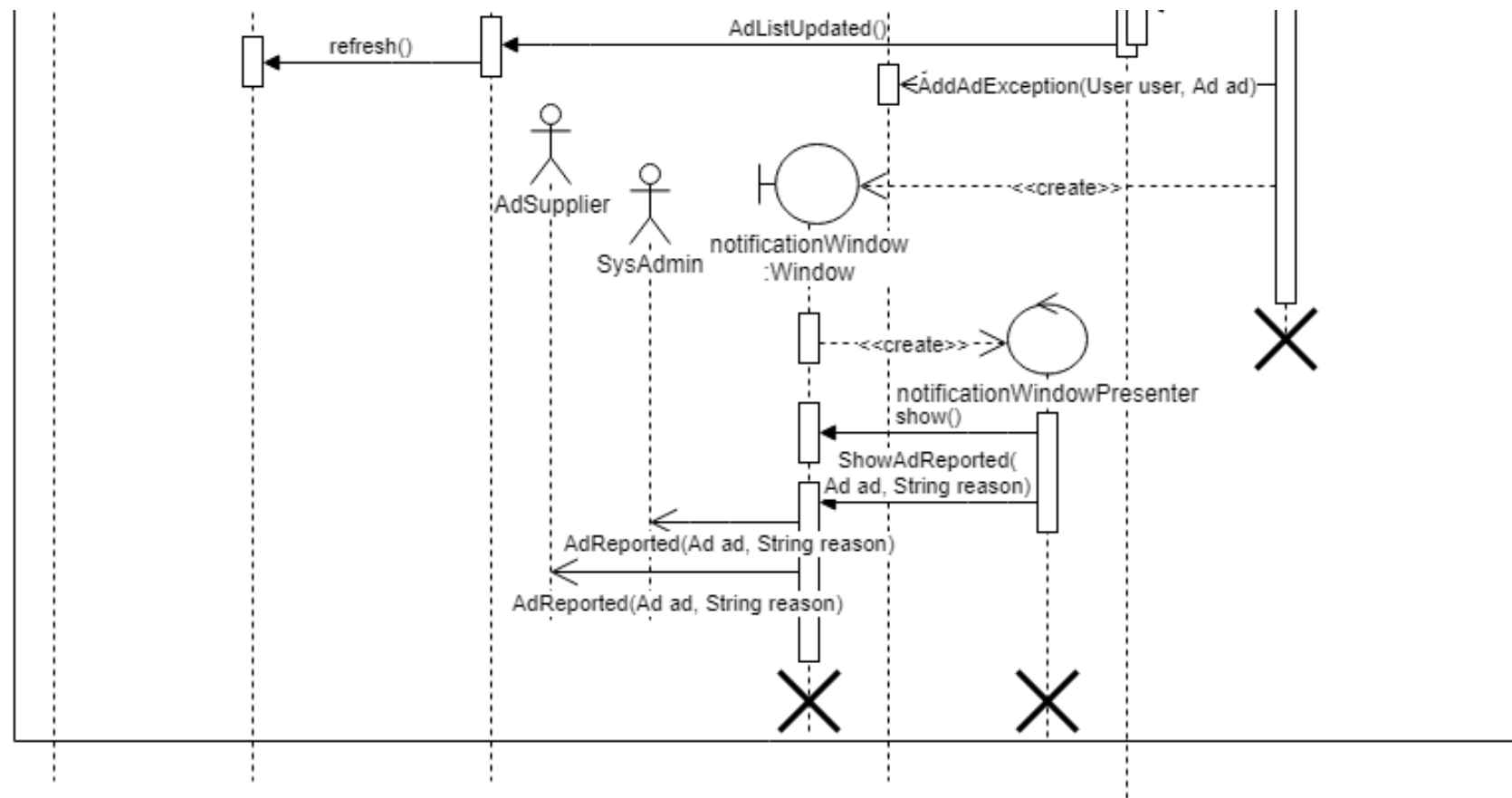
# Sekų diagrama









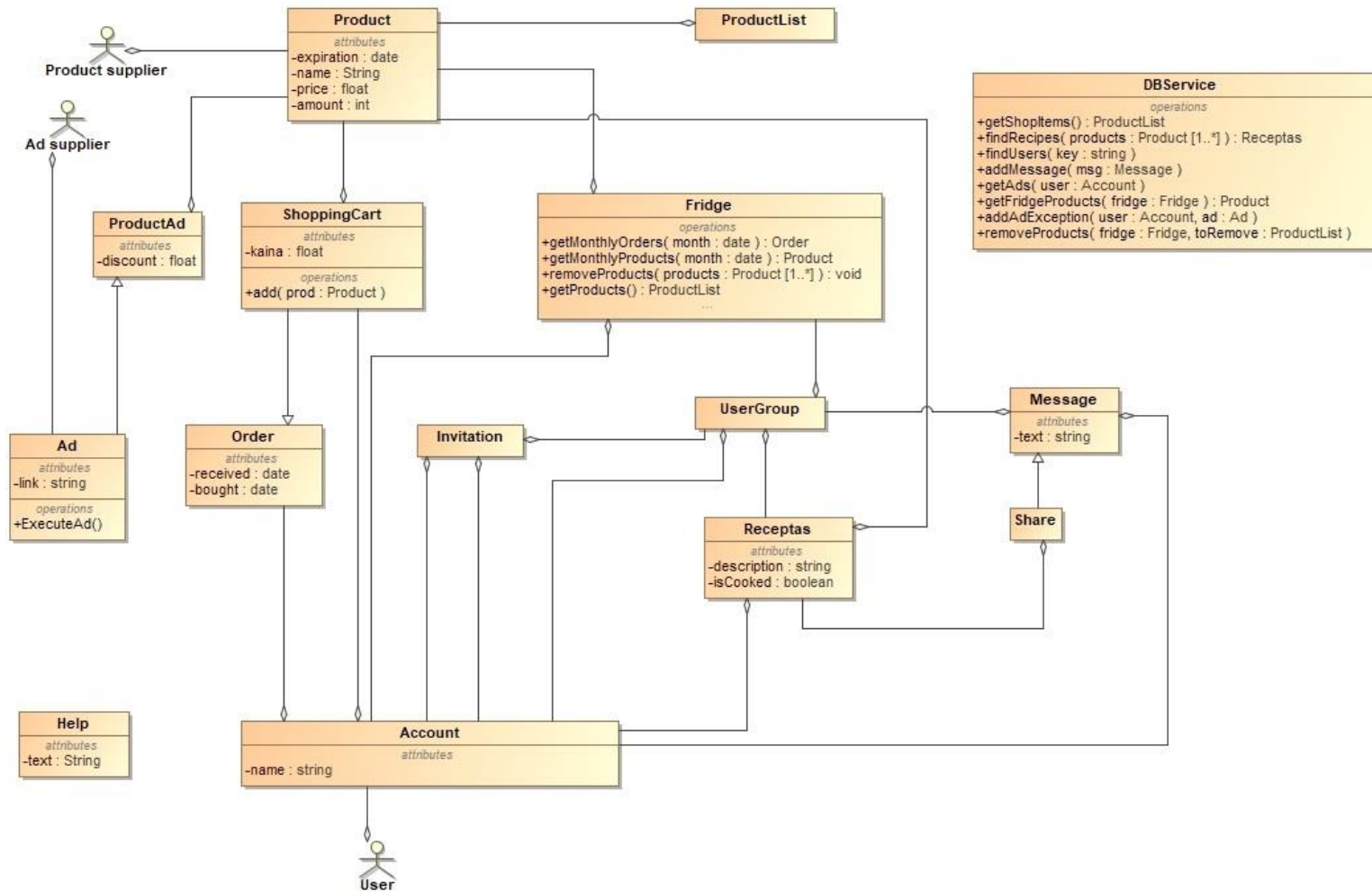


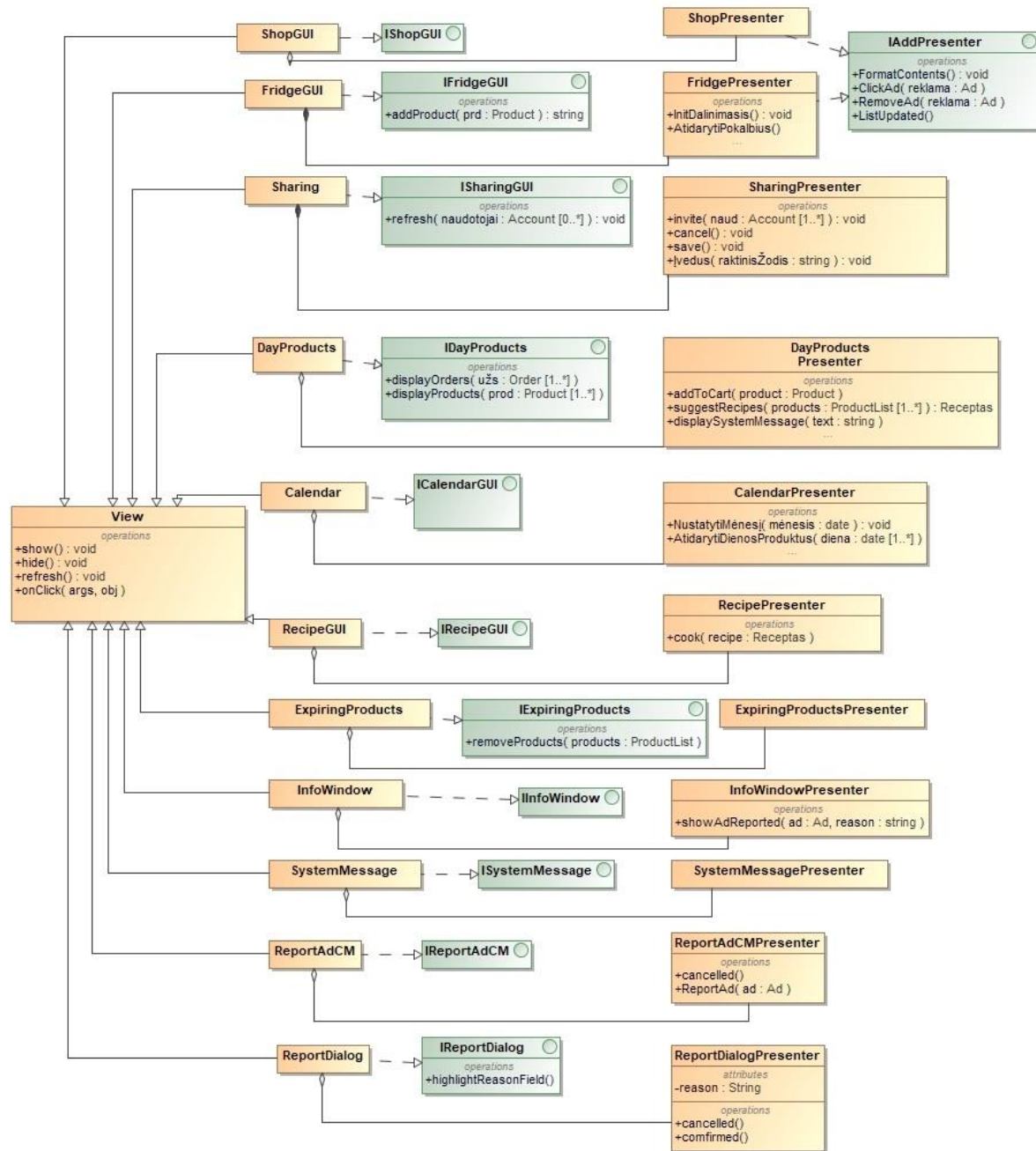


# Sistemos struktūra

- Esybių sąrašas:
  - Produktas
  - Pranešimas
  - Naudotojų grupė
  - Užsakymas
  - Tiekėjas
  - Naudotojas
  - Šaldytuvas
  - Receptas
  - Krepšelis

# Klasių diagrama





# Klasių ir reikalavimų matrica

|  | Naudotojų grupė | Pagalba | Reklama | Naudotojo paskyra | Pranešimas | Receptas | Šaldytuvas | Produktas | Aprašas | Pakvietimas | Krepšelis |
|--|-----------------|---------|---------|-------------------|------------|----------|------------|-----------|---------|-------------|-----------|
| Pranešimas dėl galiojimo laiko pabaigos                  |                 |         |         | x                 | x          |          |            | x         |         |             |           |
| Prekės užsakymas per mobiliąją programą                  |                 |         |         | x                 |            |          |            | x         |         |             | x         |
| Dalijimasis savo šaldytuvo turiniu su kitais naudotojais | x               |         |         | x                 |            |          | x          |           |         | x           |           |
| Pranešimas grupėje                                       | x               |         |         | x                 | x          | x        |            | x         |         |             |           |
| Naudotojas atsiverčia kalendorių pirkiniams peržiūrėti   |                 |         |         | x                 |            |          | x          | x         |         |             | x         |
| Naudotojas mato reklamas                                 |                 |         | x       | x                 |            |          |            |           |         |             | x         |
| Pagalba naudotojui                                       |                 | x       |         | x                 |            |          |            |           | x       |             |           |
| Pranešimas dėl klaidų                                    |                 |         |         | x                 | x          |          |            |           |         |             |           |
| Naudotojas susipažįsta su sistema                        |                 |         |         | x                 |            |          |            |           | x       |             |           |

# Projekto kritinės peržiūros rezultatai

# Dalinimasis šaldytuvu

1. Iš užduoties scenarijaus pašalintas pranešimo gavimas, kai pakviestas vartotojas priima kvietimą, ir tai perkelta į "Kvietimo Priėmimo" užduoties scenarijų
2. Sekų diagramoje pakeista, kad pakvietus vartotoją prezenteris kreipiasi į „view“ kviesdamas dialogą su pranešimu apie sėkmingą pakvietimą, papildytas užduoties tekstas
3. Sekų diagrama išversta į anglų kalbą
4. Robustiškumo diagrama pakeista, kad atitiktų naują užduoties scenarijaus tekstą

# Pakeisti projektiniai sprendimai

1. Nuspręsta iš MVC architektūros pereiti prie MVP, nes MVP labiau priimtina komandai, nes turime daugiau patirties, ir dažniau taikoma praktikoje
2. Nuspręsta programuoti anglų kalba, nes ši kalba labiau unifikuota
3. Nuspręsta vietoje daug "Event Handler" naudoti tik vieną "Event Handler" kiekvienam GUI elementui, kuris kviečia sistemos logiką ("Sanity check" rezultatas, dalyvavo programuotojai)
4. Nuspręsta naudoti NTP serverius (network time protocol), kad klientai galėtų sinchronizuotis su serveriu ir gauti pranešimus realiu laiku.

# Pokyčiai statinei struktūrai

1. ICalendarGUI papildytas dviem metodais: displayOrders(Order[]) ir displayProducts(ProductList)
2. DayProductsPresenter papildytas metodu openRecipesWindow(Recipe[]), kuris atidaro receptų langą su nurodytais receptais
3. Sudėti trūkstanti grąžinami tipai prie klasių operacijų



Testavimo planas

# Programinių vienetų testai

- Testuojamos klasės:
  - FridgePresenter
  - SharingPresenter
  - DBService
  - Fridge
  - CalendarPresenter
  - DayProductsPresenter
  - ShoppingCart

# FridgePresenter testai

```
[TestClass]
public class FridgePresenterTests
{
    [TestMethod]
    public void OnInput_UserFound_Return0()
    {
        FridgePresenter FP = new FridgePresenter(new Fridge() {nameField = "admin"});
        FP.DB = new DBService();
        int result = SV.OnInput();
        Assert.AreEqual(result, 0);
    }

    [TestMethod]
    public void OnInput_UserNotFound_Return1()
    {
        FridgePresenter FP = new FridgePresenter(new Fridge() {nameField = "*****"});
        FP.DB = new DBService();
        int result = SV.OnInput();
        Assert.AreEqual(result, 1);
    }
}
```

# SharingPresenter testai

```
[TestClass]
public class SharingPresenterTests
{
    [TestMethod]
    public void Invite_UserInvited_Return0()
    {
        IDBService DB = new DBService();
        SharingPresenter SP = new SharingPresenter(new Sharing())
        SP.Users.Add(new User() {name = "admin"});
        int result = SP.Invite();
        SP.ShowDialog("admin");
        Assert.AreNotEqual(result.Count, 0);
    }

    [TestMethod]
    public void Invite_UserNotInvited_Return1()
    {
        IDBService DB = new DBService();
        SharingPresenter SP = new SharingPresenter(new Sharing())
        SP.Users.Add(new User() {name = "admin"});
        int result = SP.Invite();
        SP.ShowDialog("admin");
        Assert.AreNotEqual(result.Count, 1);
    }
}
```

# DBService testai

```
[TestClass]
public class DBServiceTests
{
    [TestMethod]
    public void FindUser_UserFound_ReturnNot0()
    {
        IDBService DB = new DBService();
        int result = DB.FindUser("admin").count;
        Assert.AreNotEqual(result, 0);
    }

    [TestMethod]
    public void FindUser_UserNotFound_Return0()
    {
        IDBService DB = new DBService();
        int result = DB.FindUser(" *****").count; //Invalid characters ' ' and '*'
        Assert.AreEqual(result, 0);
    }

    [TestMethod]
    public void GetProduktai_ProduktaiFound_Return0()
    {
        IDBService DB = new DBService();
        int result = DB.GetProduktai(); //Connection successful, got products
        Assert.AreEqual(result, 0);
    }
}
```

# DBService testai

```
[TestClass]
public class DBServiceTest
{
    DBService DB;
    List<Product> products;

    [TestInitialize]
    public void setUp()
    {
        DB = new DBService();
        DB.knownRecipes = new List<Recipe>();
        products = new List<Product>();
        products.Add(new Product("apple"));
        products.Add( new Product("milk"));
        DB.knownRecipes.Add(new Recipe("Boiled eggs"));
    }

    [TestMethod]
    public void FindRecipesTest()
    {
        DB.knownRecipes.Add(new Recipe("Apple pie"));
        int results = DB.findRecipes(products).Count;
        Assert.AreEqual(results, 1, "Only one recipe should have been found.");
    }

    [TestMethod]
    public void NoRecipesFoundTest()
    {
        DB.knownRecipes.Add(new Recipe("French fries"));
        int results = DB.findRecipes(products).Count;
        Assert.AreEqual(results, 0, "No recipes should have been found.");
    }
}
```

# DayProductsPresenter testai

```
[TestClass]
public class DayProductsPresenterTest
{
    DayProducts view;
    DayProductsPresenter presenter;
    [TestInitialize]
    public void setUp()
    {
        view = new DayProducts();
        presenter = new DayProductsPresenter(view);
    }

    [TestMethod]
    public void AddToCartTest()
    {
        Product prod = new Product("apple"),
            prod2 = new Product("knife");
        presenter.cart.Add(prod);
        presenter.addToCart(prod2);
        Assert.AreEqual(2, presenter.cart.contents.Count, "There should be two products in the shopping car");
    }
}
```

# Sistemos užduočių testai

- Testavimo atvejai susideda iš daugybės testavimo scenarijų, dėl laiko apribojimų pateiksime tik vieną pavyzdį.
- Sistemos užduotys testuojamos rankiniu būdu (kaip priėmimo procese).
- Testavimo scenarijai gauti iš užduoties robustiškumo diagramos, kiekvienas kontrolieris turi bent vieną testavimo scenarijų.



# Bendravimas su grupe

- **Pradinės salygos:** Naudotojas prisijungęs ir yra lange „Pokalbiai“.
- TC 1: Paspausti "Siųsti pranešimą". **Laukiamas rezultatas:** Iš duomenų bazės gaunami esamos grupės nariai. Jie atvaizduojami lange "Pokalbiai". Sukuriamas naujas pranešimo objektas su tuščiu adresatų sąrašu.
- TC 2: Pasirinkti adresatą. **Laukiamas rezultatas:** Pranešimo objektas turi vienu adresatu daugiau.
- TC 2.1: Pašalinti žymėjimą ties adresatu. **Laukiamas rezultatas:** Pranešimo objektas turi vienu adresatu mažiau.
- TC 3: Paspausti "Dalintis receptais". **Laukiamas rezultatas:** Iš duomenų bazės gaunami receptai. Atidaromas langas "Receptai", kuriame pavaizduoti rasti receptai.
- TC 4 Lange "Receptai" pasirinkti receptą. **Laukiamas rezultatas:** Receptas pridedamas prie pasirinktų receptų sąrašo.

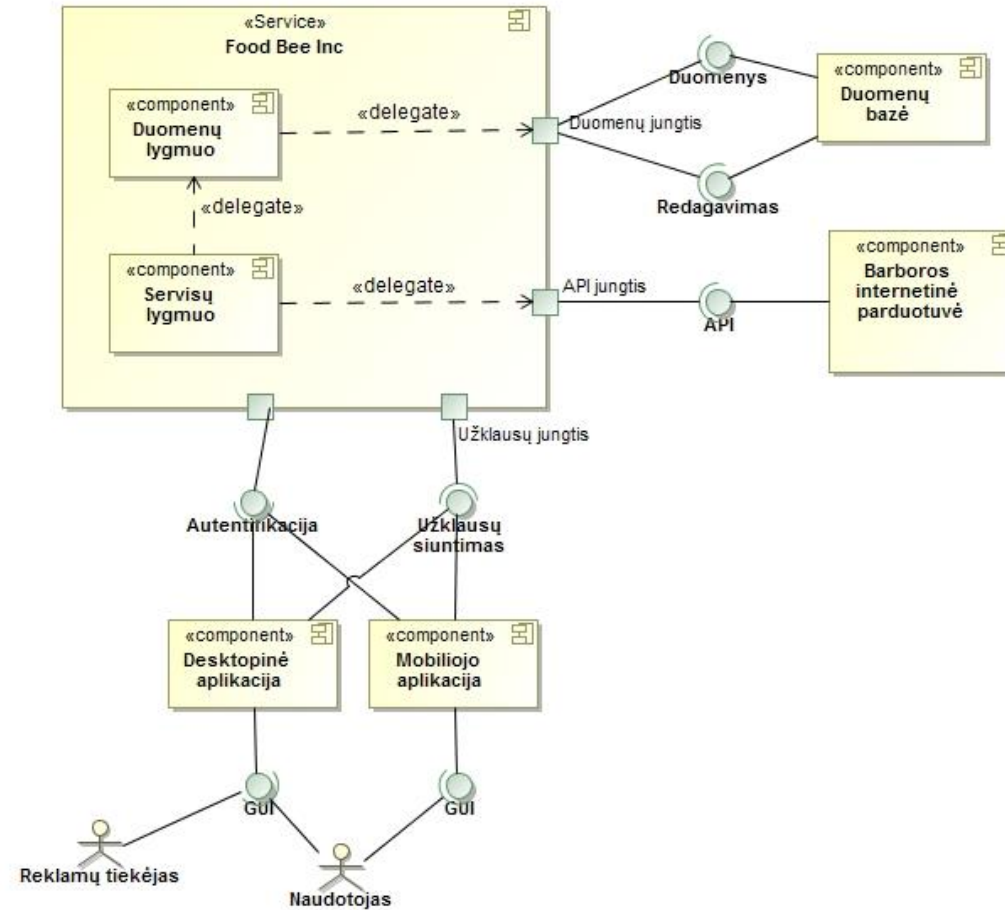
# Bendravimas su grupe

- TC 5. **Scenarijus A:** Lange "receptai" paspausti mygtuką „Patvirtinti“. **Laukiamas rezultatas:** Parinkti receptai pridedami prie dalinimosi pranešimo objekto. Grįžtama į langą „Pokalbiai“.
- **Scenarijus B:** Nepasirinkus receptų paspausti mygtuką „Patvirtinti“. **Laukiamas rezultatas:** Jokie receptai nepridedami prie pranešimo objekto. Grįžtama į langą „Pokalbiai“.
- TC 6 **Papildoma sąlyga:** Duomenų bazė pasiekiamą. **Scenarijus:** Lange "pokalbiai" spausti "siųsti". **Laukiamas rezultatas:** Duomenų bazėje atsiranda įrašas apie naują pranešimą gavėjams.
- TC 7 **Papildoma sąlyga:** Duomenų bazė nepasiekiamą. **Scenarijus:** Lange "pokalbiai" spausti "siųsti". **Laukiamas rezultatas:** atidaromas Sisteminio pranešimo langas.
- TC 8.1: Sisteminio pranešimo lange spausti „Siųsti iš naujo“. **Laukiamas rezultatas:** Pranešimą bandoma siųsti iš naujo.
- TC 8.2: Sisteminio pranešimo lange spausti „atšaukti“. **Laukiamas rezultatas:** Uždaromas sisteminio pranešimo langas.

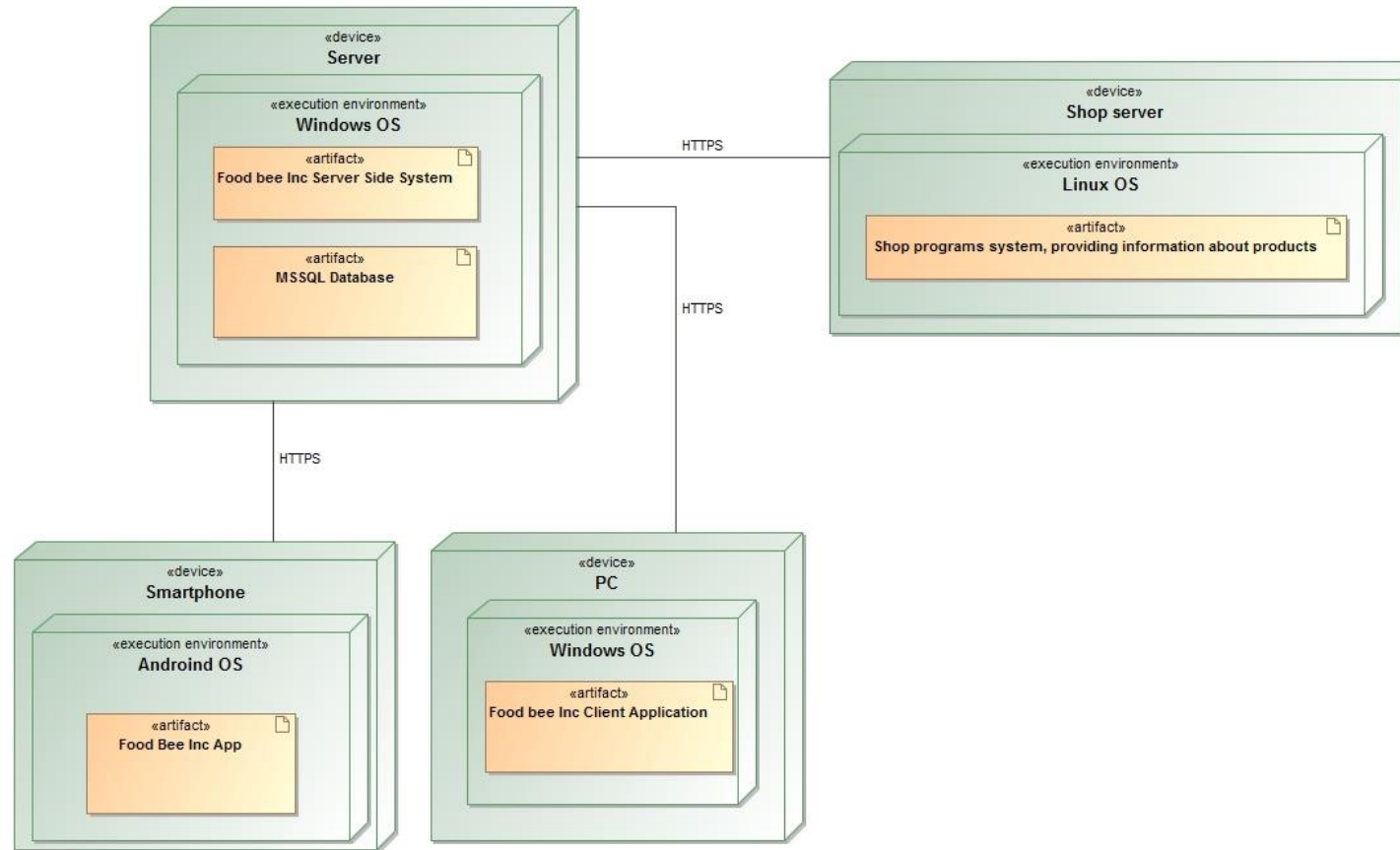
# Sistemos techninė architektūra

- Nuspręsta sistemos techninę architektūrą pavaizduoti dviem pjūviais:
  - Fiziniu,
  - Kūrimo.
- Fizinis pjūvis - programų sistemos komponentų išsidėstymas tinkle ir kokiais protokolais bendraujama tarpusavyje.
- Kūrimo pjūvis - loginis sąryšis tarp komponentų ir kaip su jais sąveikauja agentai.
- Numatoma kliento-serverio architektūra.

# Sistemos komponentai



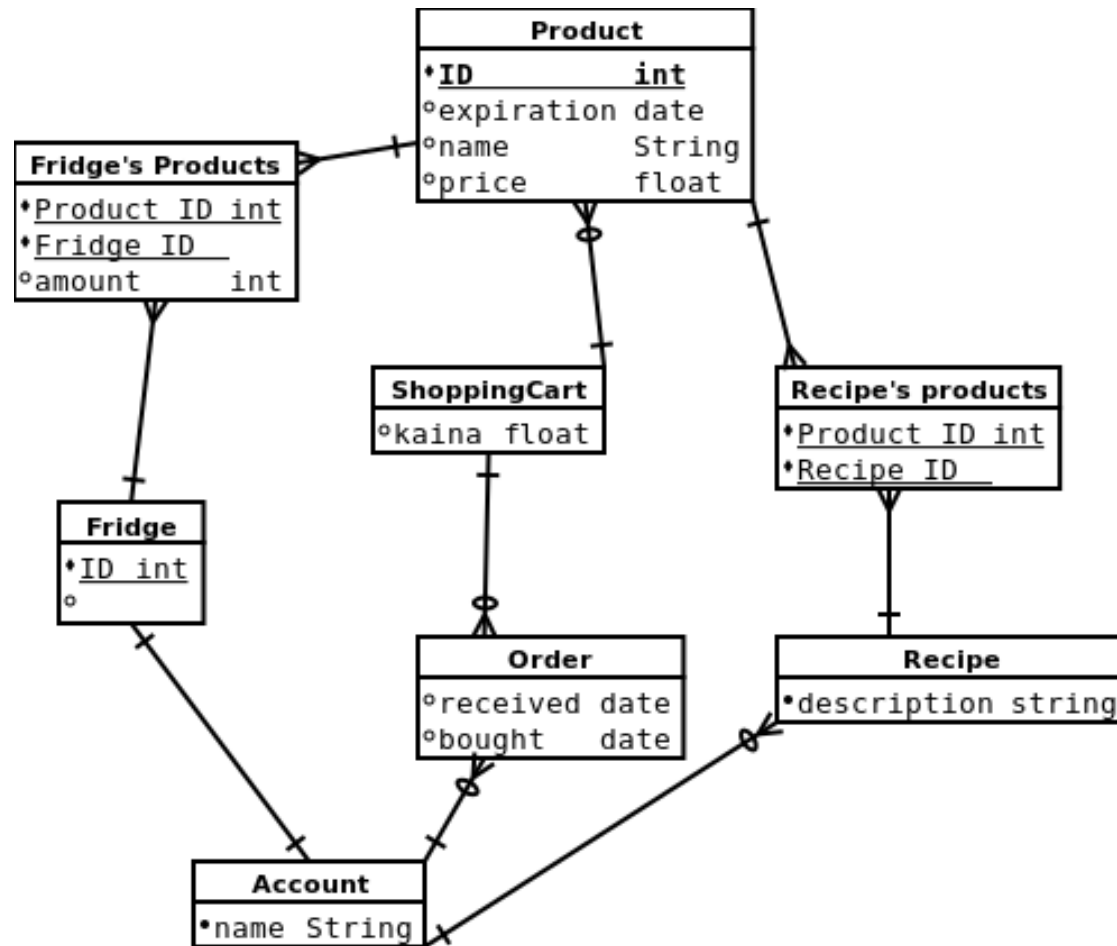
# Fizinis pjūvis



Sistemas realizacija

# Duomenų bazės schema

- Sukurta pradinė duomenų bazės schemas versija.
- Sukurtos kelios lentelės ryšiui N-N (daug su daug) įgyvendinti.
- Dalies klasių informacija netalpinama duomenų bazėje (pvz., pagalbos pranešimai)

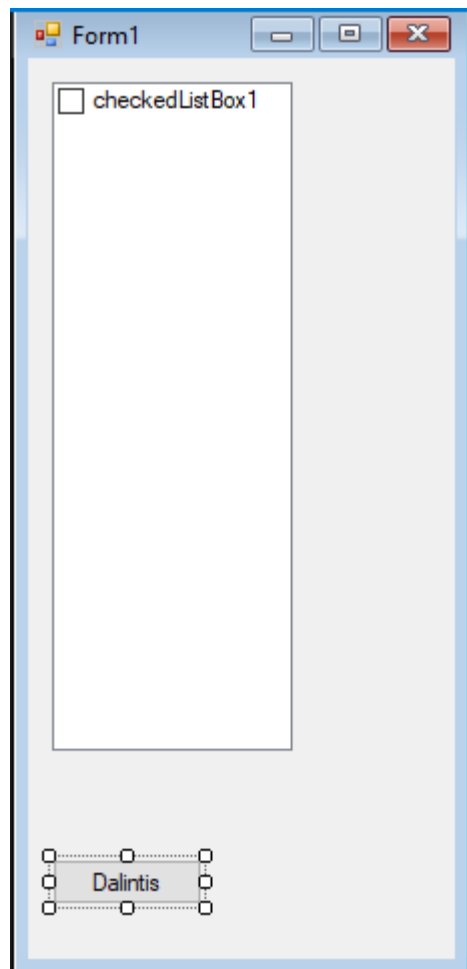


# Pradiniai programų kodai

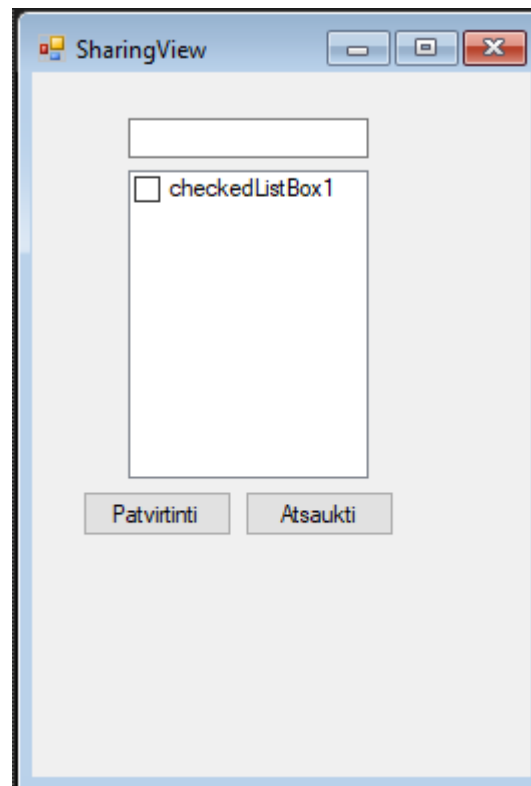
- Sukurtas prototipas, kuriame įgyvendinta šaldytuvo dalinimosi užduotis.
- Prototipas testuojamas anksčiau pavaizduotais vienetų testais.
- Dalinai įgyvendintos klasės:
  - DBService
  - Account
  - Invitation
  - Product
  - SharingView
  - FridgeView
  - SharingPresenter
  - FridgePresenter



# FridgeView ir SharingView



A screenshot of a Windows form titled "Form1". The form has a light gray background. At the top, there is a standard Windows window title bar with minimize, maximize, and close buttons. Below the title bar, there is a white rectangular area containing a checkbox labeled "checkedListBox1". At the bottom of the form, there is a button labeled "Dalintis" with a dashed border and small square handles, indicating it is in a design or move state.



A screenshot of a Windows form titled "SharingView". The form has a light gray background. At the top, there is a standard Windows window title bar with minimize, maximize, and close buttons. Below the title bar, there is a white rectangular area containing a checkbox labeled "checkedListBox1". Below this area, there are two buttons: "Patvirtinti" and "Atsaukti".

# Šaltiniai

- Projekto talpykla (<https://github.com/Tristanas/PSI2-Food-Bee-Inc>)
- Projekto šaltinių dokumentas:  
(<https://github.com/Tristanas/PSI2-Food-Bee--nc/blob/master/README.md>)

Klausimai?

Ačīū už dėmesį!