

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Дальневосточный федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
Департамент программной инженерии и искусственного интеллекта**

РАЗРАБОТКА СИСТЕМЫ, ОСНОВАННОЙ НА ЗНАНИЯХ, ДЛЯ

ПРЕДМЕТНОЙ ОБЛАСТИ

«ПРОЕКТИРОВАНИЕ РАСПИСАНИЯ ПАССАЖИРСКИХ АВИАПЕРЕВОЗОК»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

по дисциплине «Методы и технологии интеллектуализации программных систем» по образовательной программе подготовки бакалавров по направлению 09.03.04 «Программная инженерия»

Выполнили студенты гр. Б9120-09.03.04прогин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Воробьева К.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузнецов Е.А.

Проверила д.т.н., профессор Артемьева И.Л.

\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

(подпись)

г. Владивосток  
2024

Оглавление

[Введение 3](#_Toc163956999)

[1 Анализ предметной области 4](#_Toc163957000)

[1.1 Анализ множества задач профессиональной деятельности 4](#_Toc163957001)

[1.2 Анализ знаний предметной области 5](#_Toc163957002)

[2 Построение модели предметной области 6](#_Toc163957003)

[2.1 Онтология знаний 6](#_Toc163957004)

[2.2 Онтология действительности 8](#_Toc163957005)

[2.3 Онтологические соглашения 9](#_Toc163957006)

[3 Проект системы, основанной на знаниях 10](#_Toc163957007)

[3.1 Архитектура системы 10](#_Toc163957008)

[3.2 Use-case диаграмма 12](#_Toc163957009)

[3.3 Проект базы данных для хранения данных 12](#_Toc163957010)

[3.4 Проект базы данных для хранения исходных данных 13](#_Toc163957011)

[3.5 Проектирование интерфейса системы, основанной на знаниях 14](#_Toc163957012)

[3.5.1 Определение частичного порядка на множестве терминов области знаний 14](#_Toc163957013)

[3.5.2 Сценарий диалога с экспертом при формировании базы знаний 14](#_Toc163957014)

[3.6 Проект интерфейса системы ввода исходных данных 17](#_Toc163957015)

[3.6.1 Определение частичного порядка на множестве терминов области знаний 17](#_Toc163957016)

[3.6.2 Сценарий диалога при вводе исходных данных 17](#_Toc163957017)

[3.7 Проект решателя задач 19](#_Toc163957018)

[3.7.1 Описание работы решателя задач 20](#_Toc163957019)

[3.7.2 Сценарий диалога с решателем задач 20](#_Toc163957020)

[4 Реализация и тестирование 21](#_Toc163957021)

[4.1 Средства разработки 21](#_Toc163957022)

[4.2 Тестирование 22](#_Toc163957023)

[Заключение 27](#_Toc163957024)

[Список литературы 28](#_Toc163957025)

# Введение

Система, основанная на знаниях – сложный программный комплекс, аккумулирующий знания специалистов в конкретных предметных областях, который помогает менее квалифицированным пользователям решать свои задачи.

Основными компонентами системы, основанной на знаниях, являются редактор знаний, решатель задач, подсистема ввода исходных знаний.

Одним из классов таких систем является задача проектирования. Она состоит в нахождении проекта (устройства) объекта задачи по характеристикам его частей и критериям отбора проектов. Знания предметной области определяют возможности компоновки составных частей проекта, а также выявление множества ограничений на проект.

В последние годы, в связи с автоматизацией различных сфер деятельности, рынок IT-технологий растет стремительно. Ранее программное обеспечение использовалось лишь в ограниченном числе сфер, но сегодня практически каждая компания, предприятие и организация, а также большинство людей нуждаются в разнообразных приложениях, которые облегчают их повседневную жизнь и работу.

Таким образом, цель лабораторной работы – разработка проекта системы, основанной на знаниях.

Для достижения поставленной цели необходимо решить следующие задачи:

• проанализировать предметную область;

• разработать модель предметной области;

• разработать проект системы и пользовательского интерфейса;

* реализовать программное средство и провести тестирование.

# Анализ предметной области

## Анализ множества задач профессиональной деятельности

В данной предметной области составителем расписания авиаперевозок (профессионалом предметной области) решается задача проектирования расписания авиаперевозки пассажиров. Объектом этой задачи является расписание авиаперевозки.

Под расписанием авиаперевозки пассажиров понимается сопоставление каждому полету некоторого самолета, маршрута полета и времени полета. Результатом решения задачи является проект расписания авиаперевозки пассажиров.

При проектировании расписания авиаперевозки пассажиров учитывается тип самолета (от которого зависит возможная скорость и дальность полета), время полета и маршрут, по которому следует самолет.

Под самолетом (воздушным судном) понимается некоторый уникальный идентификатор самолета, следующего по маршруту.

Под маршрутом понимается пара: город отправления и город прибытия воздушного судна, при этом город отправления не может совпадать с городом прибытия.

Под временем полета понимается пара: начало и конец полета. Под началом и концом полета понимаются размерные значения с размерностью «месяцы» ✕ «дни» ✕ «часы» ✕ «минуты». Под минутами понимаются целые числа от 0 до 59, под часами понимаются целые числа от 0 до 23. Под днями понимаются целые числа от 1 до 31, под месяцами понимаются целые числа от 1 до 12.

Результатом решения задачи является проект расписания авиаперевозки пассажиров. Под проектом расписания авиаперевозки пассажиров понимается задание такого сопоставления, которое будет согласовано с требованиями, предъявляемыми к расписанию авиаперевозки пассажиров.

## Анализ знаний предметной области

В этом разделе выполняется анализ знаний предметной области, формулируются законы предметной области на естественном языке.

**Ограничения целостности области знаний:**

* Город отправления и город прибытия для маршрута не совпадают.
* Городов должно быть не меньше двух.

**Ограничения целостности области действительности:**

* Время начала полета меньше времени конца полета.
* В одно и тоже время не может быть полетов с одинаковым маршрутом и самолетом.
* Должен существовать хотя бы один самолет.

**Ограничения связей области знаний и области действительности:**

* Время полета зависит от дальности маршрута и скорости самолета.
* Тип самолета характеризуется скоростью самолета и дальностью полета.
* Для каждого самолета существует хотя бы один маршрут.

**Утверждения, задающие дальности полета:**

* Если тип самолета – дальнемагистральное воздушное судно, то его дальность полета составляет свыше 6000 км.
* Если тип самолета – среднемагистральное воздушное судно, то его дальность полета составляет от 1000 км до 6000 км.
* Если тип самолета – самолет местных авиалиний, то его дальность полета составляет от 400 км до 1000 км.

**Утверждения, задающие скорость самолета:**

* Если тип самолета – дозвуковое воздушное судно, то его крейсерская скорость от 600 км/ч до 800 км/ч.
* Если тип самолета – трансзвуковое воздушное судно, то его крейсерская скорость от 800 км/ч до 1200 км/ч.
* Если тип самолета – сверхзвуковое воздушное судно, то его крейсерская скорость от 1200 км/ч до 2000 км/ч.

# Построение модели предметной области

## Онтология знаний

**Определение вспомогательных терминов:**

* 1. множества значений ≡ ({} N) ∪ ([ ]I) ∪ ([ ]R)

термин "множества значений" обозначает объединение множеств всех подмножеств имен (области качественных значений), целочисленных и вещественных интервалов (области количественных значений)

* 1. скорость движения ≡ R[0, 2000]

термин "скорость движения" эквивалентен значению крейсерской скорости воздушного судна км/ч

* 1. дальность полета ≡ R(0, ∞)

термин "дальность полета" означает дальность полета воздушного судна в км

* 1. месяцы ≡ I[1, 12]

термин "месяцы" эквивалентен номеру календарного месяца

* 1. дни ≡ I[1, 31]

термин "дни" эквивалентен числу календарного месяца

* 1. часы ≡ I[0, 23]

термин "часы" эквивалентен единице времени – часам

* 1. минуты ≡ I[0, 59]

термин "минуты" эквивалентен единице времени – минутам

* 1. время ≡ ✕ месяцы, дни, часы, минуты

термин "время" эквивалентен времени в формате месяцы.дни.часы.минуты

* 1. типы самолетов ≡ {дальнемагистральное дозвуковое воздушное судно, дальнемагистральное трансзвуковое воздушное судно, дальнемагистральное сверхзвуковое воздушное судно, среднемагистральное дозвуковое воздушное судно, среднемагистральное трансзвуковое воздушное судно, среднемагистральное сверхзвуковое воздушное судно, дозвуковое воздушное судно местных авиалиний, трансзвуковое воздушное судно местных авиалиний, сверхзвуковое воздушное судно местных авиалиний}

термин "типы самолетам" эквивалентен типам пассажирских самолетов

**Описание терминов для описания знаний:**

1. сорт города: {} N \ {∅}

термин "города" обозначает конечное множество городов

1. сорт маршруты: {} {(v: города ⇑ 2) | π(1, v) ≠ π(2, v)}

термин "маршруты " обозначает конечное множество маршрутов: город отправления и город прибытия

1. сорт дальность маршрута: маршруты → R(0, ∞)

термин "дальность маршрута" обозначает функцию, которая сопоставляет каждому маршруту его протяжённость

1. сорт скорость движения для самолета определенного типа: типы самолетов → {(v: скорость движения ⇑ 2) | π(1, v) ≤ π(2, v)}

термин "скорость движения для самолета определенного типа" обозначает функцию, которая сопоставляет каждому самолету определённого типа интервал допустимой скорости движения

1. сорт дальность полета для самолета определенного типа: типы самолетов→ {(v: дальность полета ⇑ 2) | π(1, v) ≤ π(2, v)}

термин "дальность полета для самолета определенного типа" обозначает функцию, которая сопоставляет каждому самолету определённого типа интервал допустимой дальности полета

## Онтология действительности

1. сорт самолеты: {} N \ {∅}

термин "самолеты" обозначает конечное множество идентификаторов самолетов

1. сорт тип самолета: самолеты → типы самолетов

термин "тип самолета" обозначает функцию, которая сопоставляет каждому самолету его тип

1. сорт время полета: {(t: время ⇑ 2) | π(1, π(1, t)) \* 43830 + π(2, π(1, t)) \* 1440 + π(3, π(1, t)) \* 60 + π(4, π(1, t)) < π(1, π(2, t)) \* 43830 + π(2, π(2, t)) \* 1440 + π(3, π(2, t)) \* 60 + π(4, π(2, t))}

термин "время полета" обозначает время полета, при этом время начала полета не должно быть больше времени конца полета

1. сорт скорость самолета: самолеты → скорость движения

термин "скорость самолета" обозначает функцию, которая сопоставляет каждому самолету скорость движения

1. сорт дальность полета самолета: самолеты → дальность маршрута

термин "дальность маршрута самолета" обозначает функцию, которая сопоставляет каждому самолету дальность маршрута

1. сорт маршруты самолетов: {} маршруты \ {∅}

термин "маршруты самолетов" обозначает конечное множество маршрутов самолетов

1. сорт полеты: {} ✕ тип самолета, маршруты, время полета

термин "полеты" обозначает конечное множество полетов самолетов

1. сорт расписание авиарейсов самолетов: {} ✕ самолеты, маршруты самолетов, время полета

термин "расписание авиарейсов самолетов" обозначает расписание движения самолетов

## Онтологические соглашения

**Ограничения для самолета определенного типа:**

(∀ v ∊ расписание авиарейсов самолетов):

* скорость самолета (π(1, v)) ≥ (π(1, скорость движения для самолета определенного типа (тип самолета(π(1, v))))) & скорость самолета (π(1, v)) ≤ (π(2, скорость движения для самолета определенного типа (тип самолета(π(1, v))))) – скорость самолета должна соответствовать допустимому интервалу значений скорости движения для самолета определенного типа.
* дальность полета (π(1, v)) ≥ (π(1, дальность полета для самолета определенного типа (тип самолета(π(1, v))))) & дальность полета (π(1, v)) ≤ (π(2, дальность полета для самолета определенного типа (тип самолета(π(1, v))))) – дальность полета должна соответствовать допустимому интервалу значений дальности полета для самолета определенного типа.

**Ограничения на время полета:**

* (∀ v1, v2  ∊ расписание авиарейсов самолетов: v1≠ v2 ): (π(1, v1) = π (1, v2) & π(2, v1) = π (2, v2)) => π(3, v1) ≠ π (3, v2) – не может быть нескольких полетов с одинаковым маршрутом в одно и то же время.
* (∀ v ∊ расписание авиарейсов самолетов): дальность полета(π(2, v)) / скорость самолета(π(1, v)) = (π(1, π(2, π(3, v))) \* 43830 + (π(2, π(2, π(3, v))) \* 1440 + (π(3, π(2, π(3, v))) \* 60 + π(4, π(2, π(3, v)))) – (π(1, π(1, π(3, v))) \* 43830 + (π(2, π(1, π(3, v))) \* 1440 + (π(3, π(1, π(3, v))) \* 60 + π(4, π(1, π(3, v)))) – время полета зависит от дальности маршрута и скорости самолета.

**Ограничения на маршрут****:**

(∀ v ∊ расписание авиарейсов самолетов):

* π(1, π(1, v)) = π(1, π(2, v)) & π(3, π(1, v)) = π(3, π(2, v)) => π(2, π(1, v) = π(2, π(2, v)) – один и тот же самолет не может участвовать на нескольких маршрутах одновременно.
* π(1, π(2, v)) ≠ π(2, π(2, v)) – город отправления и город прибытия различны.
* µ(города) > 1 – городов должно быть не меньше двух.

# Проект системы, основанной на знаниях

## Архитектура системы

Ниже представлена контекстная диаграмма верхнего уровня системы (Рисунок 1). На вход системы принимаются знания ПО и исходные данные. Результатом работы системы является результат решения задачи проектирования расписания пассажирских авиаперевозок в понятной для пользователя форме.



Рисунок 1 – Контекстная диаграмма системы

Далее представлена архитектурно-контекстная диаграмма системы (Рисунок 2). На диаграмме представлены составные части системы: решатель задач, редактор знаний, редактор исходных данных, база знаний, база исходных данных, интерфейс ввода/вывода.

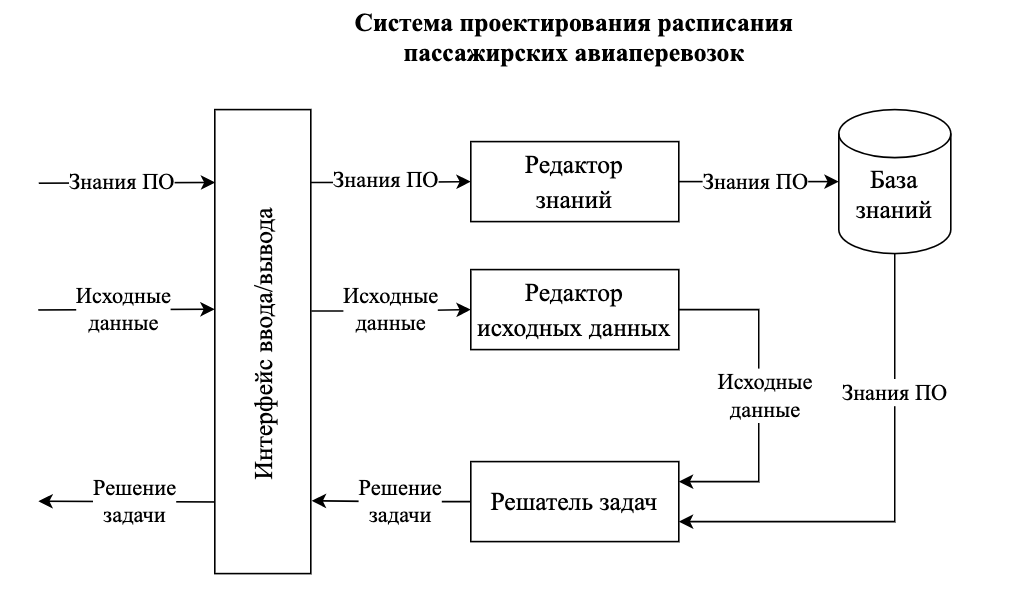


Рисунок 2 – Архитектурно-контекстная диаграмма системы

Редактор знаний используется для создания и редактирования базы знаний. Специалисты ПО вносят в редактор свои знания о ней. База знаний предназначена для хранения знаний ПО, внесённых с помощью редактора знаний.

Редактор исходных данных используется пользователем для ввода и редактирования исходных данных в базу исходных данных.

С помощью пользовательского интерфейса в систему вводятся знания ПО и исходные данные пользователя для решения задач. Здесь входные данные проходят проверки на корректность и целостность. Также с помощью пользовательского интерфейса выводятся результаты решения задач в понятной пользователю форме.

Решатель задач выдаёт результат решения задач на основе пользовательских данных.

## Use-case диаграмма

Диаграмма use-case представлена на рисунке 3. Эксперт может использовать систему для создания, редактирования и проверки базы знаний. Пользователь может использовать систему для ввода исходных данных и решения задач с помощью решателя задач.

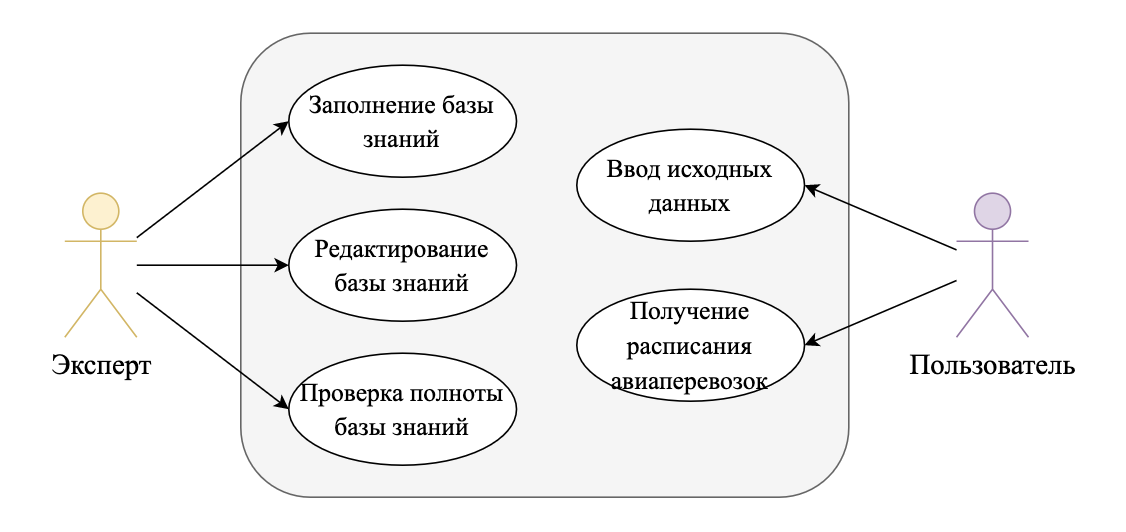


Рисунок 3 – Use-case диаграмма

## Проект базы данных для хранения данных

На рисунке 4 представлена схема базы данных для хранения знаний.

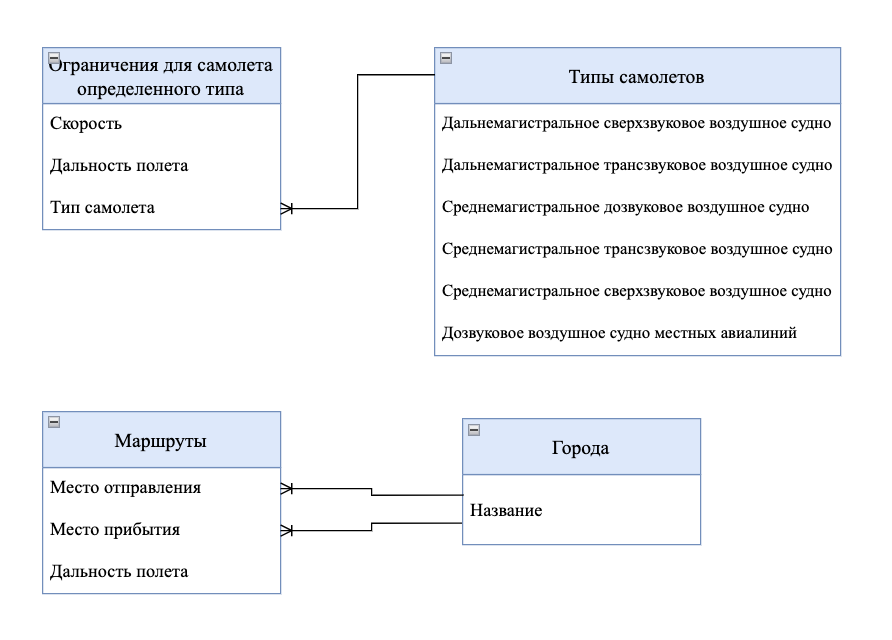


Рисунок 4 – Схема базы данных для хранения знаний

## Проект базы данных для хранения исходных данных

На рисунке 5 представлена схема базы данных для хранения исходных данных.

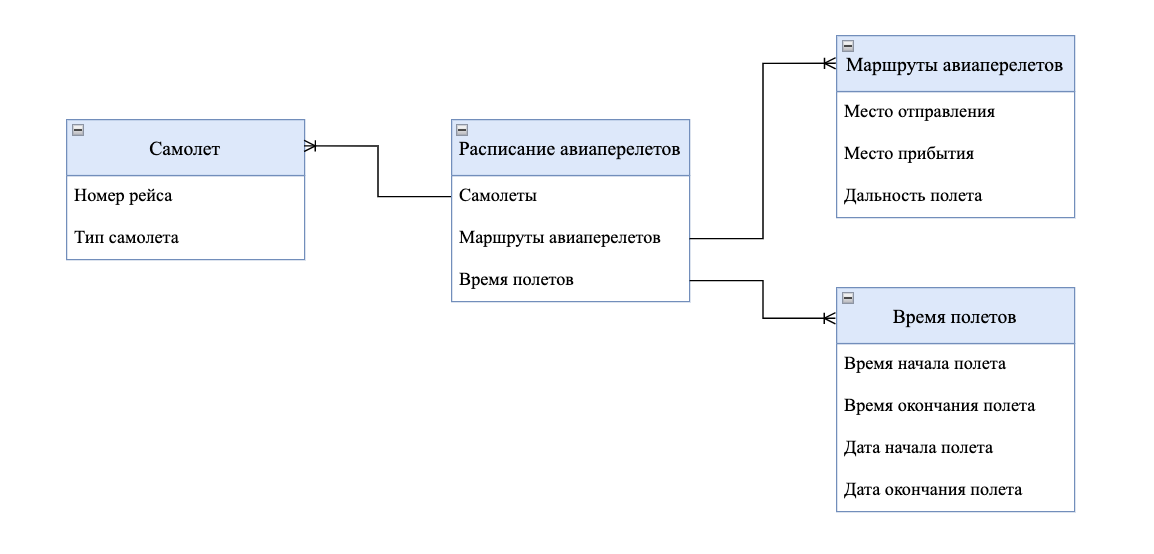


Рисунок 5 – Схема базы данных для хранения исходных данных

## Проектирование интерфейса системы, основанной на знаниях

### Определение частичного порядка на множестве терминов области знаний

На терминах из области знаний вводится структура частичного порядка по отношению зависимости. Эту структуру можно описать графом (Рисунок 6).

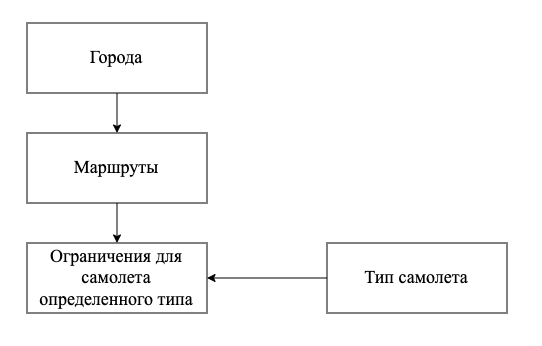


Рисунок 6 – Граф зависимостей терминов области знаний

В вершинах графа находятся термины, между двумя вершинами существует ребро от первой ко второй, если вторая вершина определяется через первую.

### Сценарий диалога с экспертом при формировании базы знаний

Далее представлен возможный сценарий диалога с экспертом в редакторе базы знаний.

На рисунке 7 представлен возможный интерфейс ввода городов.

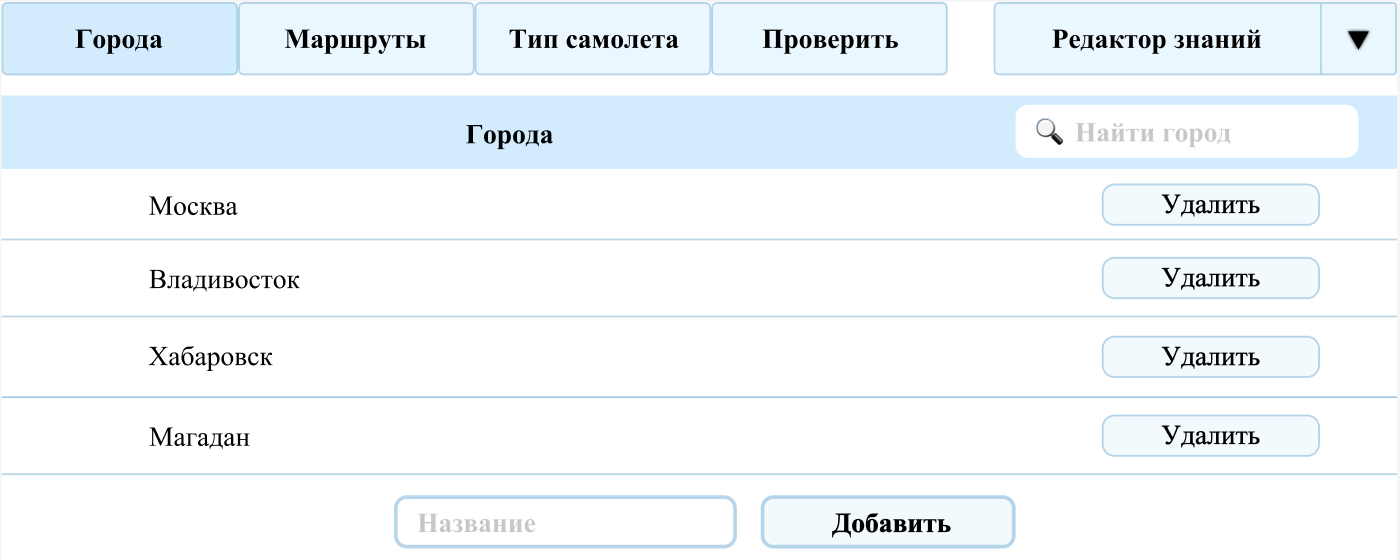


Рисунок 7 – Проект интерфейса добавления городов

Для добавления города необходимо ввести название города в текстовое поле, затем нажать кнопку добавить. Новый город не будет добавлен, если поле не будет заполнено. Для удаления города необходимо нажать на кнопку «Удалить» рядом с соответствующей строкой.

Для редактирования записей необходимо изменить требуемое поле, все изменения будут автоматически сохранены.

На рисунке 8 представлен возможный интерфейс ввода маршрута полета.



Рисунок 8 – Проект интерфейса добавления маршрута полета

Для добавления маршрута полета необходимо выбрать город отправления и город прибытия из выпадающего меню, указать в текстовом поле дальность маршрута, после чего необходимо нажать кнопку «Добавить». В случае, если хотя бы одно из полей не было заполнено, маршрут добавлен не будет. Также предусмотрена возможность удаления маршрута, для этого необходимо нажать на кнопку «Удалить» напротив соответствующей строки маршрута, после чего запись будет немедленно удалена.

Для редактирования записей необходимо изменить требуемое поле, все изменения будут автоматически сохранены.

На рисунке 9 представлен возможный интерфейс ввода ограничений для типа самолетов.



Рисунок 9 – Проект интерфейса ввода ограничений для типа самолета

Для добавления ограничений для типа самолета необходимо выбрать из всплывающего меню тип самолета, ввести начало и окончание интервала допустимых значений для крейсерской скорости самолета и дальности полета в текстовые поля и затем нажать на кнопку «Добавить». Новое ограничение не будет добавлено, если хотя бы одно поле не будет заполнено. Для удаления ограничения необходимо нажать на кнопку «Удалить» рядом с соответствующей строкой.

Для редактирования записей необходимо изменить требуемое поле, все изменения будут автоматически сохранены.

Проверка полноты задания базы знаний экспертом происходит при нажатии на кнопку «Проверить на полноту» (Рисунок 10). Значения полей не могут быть пустыми, иначе данные не будут сохранены, а эксперт получит сообщение об ошибке.

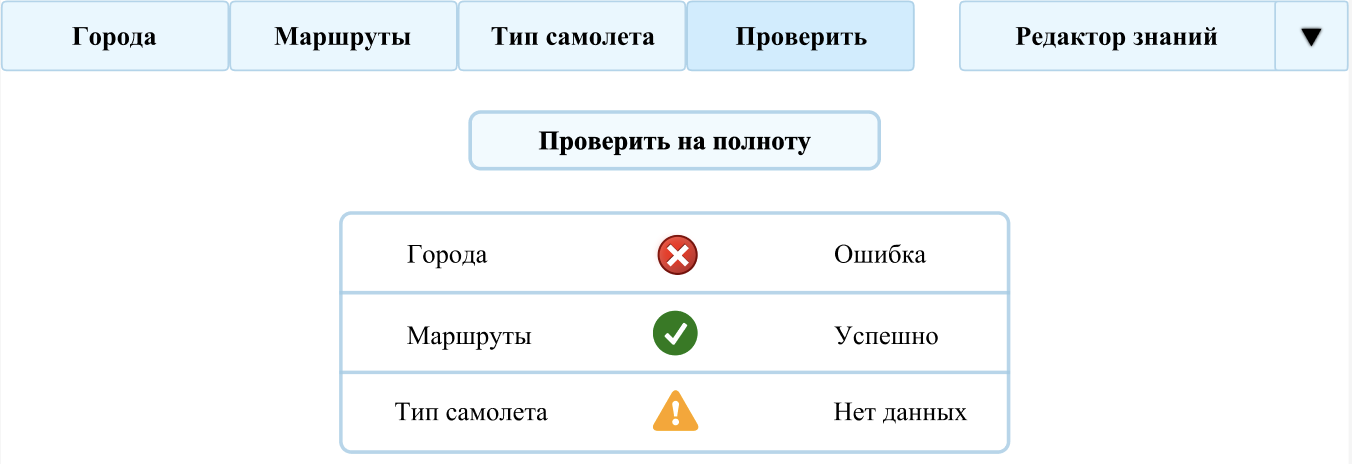


Рисунок 10 – Проект интерфейса проверки полноты

## Проект интерфейса системы ввода исходных данных

### Определение частичного порядка на множестве терминов области знаний

На терминах из области действительности вводится структура частичного порядка по отношению зависимости. Эту структуру можно описать графом, который приведен ниже на рисунке 11. В вершинах графа находятся термины, между двумя вершинами существует ребро от первой ко второй, если вторая вершина определяется через первую.

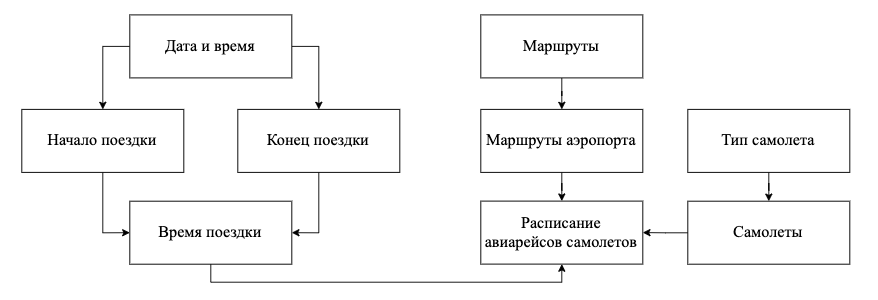


Рисунок 11 – Граф зависимостей терминов области действительности

### Сценарий диалога при вводе исходных данных

На рисунке 12 представлен возможный интерфейс ввода самолетов.

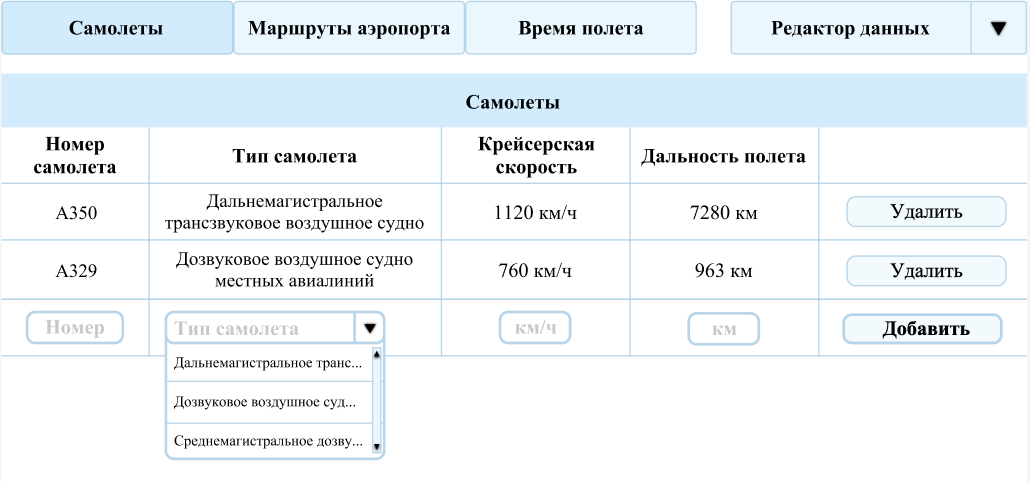


Рисунок 12 – Проект интерфейса ввода самолетов

Для добавления самолета необходимо ввести его номер, дальность полета и крейсерскую скорость самолета в текстовые поля и выбрать из всплывающего меню тип самолета, затем нажать на кнопку «Добавить». Новый самолет не будет добавлен, если хотя бы одно поле не будет заполнено. Для удаления самолета необходимо нажать на кнопку «Удалить» рядом с соответствующей строкой.

На рисунке 13 представлен возможный интерфейс ввода маршрутов аэропорта.

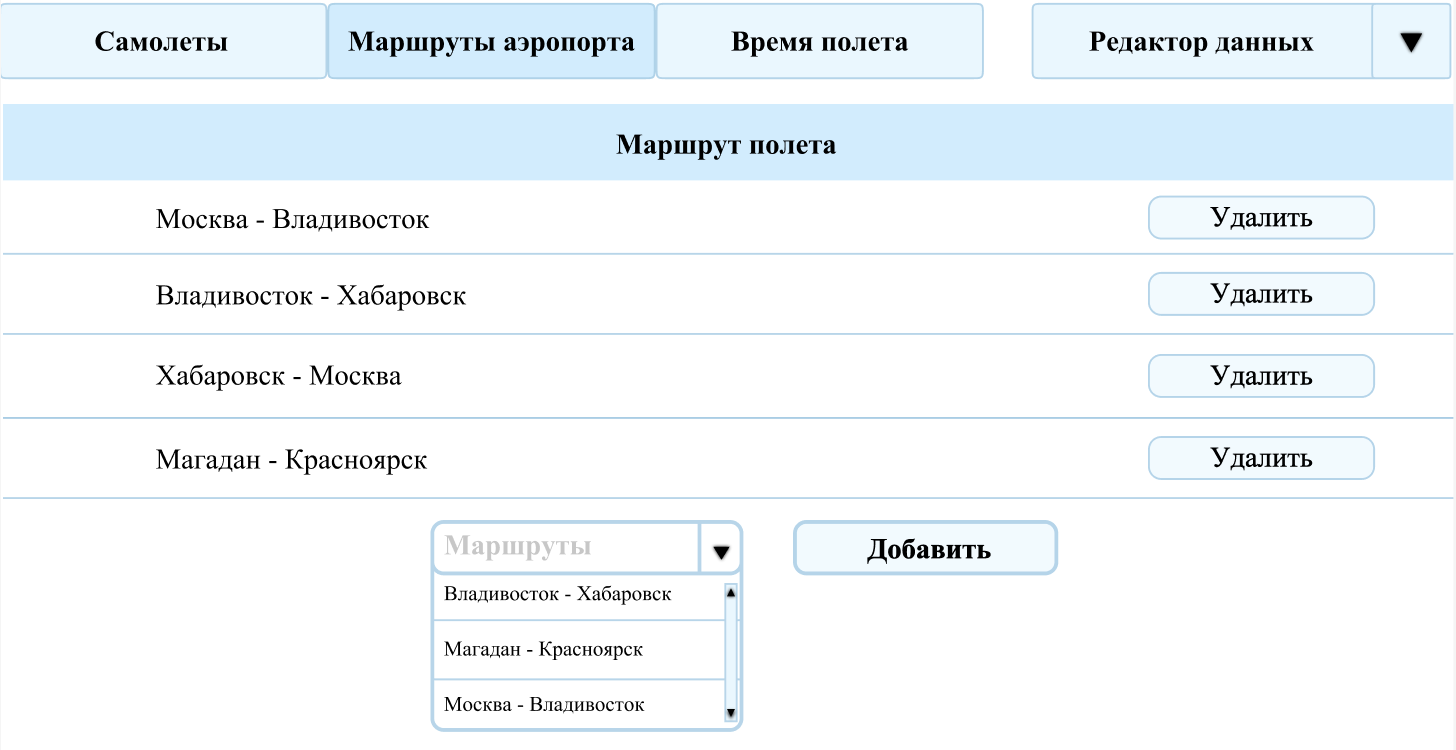


Рисунок 13 – Проект интерфейса ввода маршрутов аэропорта

Для добавления маршрута аэропорта необходимо выбрать из всплывающего меню маршрут, затем нажать на кнопку «Добавить». Новый маршрут не будет добавлен, если поле не будет заполнено. Для удаления маршрута необходимо нажать на кнопку «Удалить» рядом с соответствующей строкой.

На рисунке 14 представлен возможный интерфейс ввода времени полета.



Рисунок 14 – Проект интерфейса ввода времени полета

Для добавления времени полета необходимо выбрать из всплывающего меню маршрут и ввести время и дату полета в текстовые поля, затем нажать на кнопку «Добавить». Новое время полета не будет добавлено, если хотя бы одно поле не будет заполнено. Для удаления времени полета необходимо нажать на кнопку «Удалить» рядом с соответствующей строкой

## Проект решателя задач

Ниже приведено описание работы решателя задач, выполняющего функцию проектирования расписания пассажирских авиаперевозок, используя знания из ПО и исходные данные, введенные пользователем.

### Описание работы решателя задач

Для каждого маршрута и соответствующего времени полета выбирается самолет из числа тех, что были добавлены пользователем. Получившаяся тройка (самолет, маршрут, время полета) проверяется на наличие противоречий с онтологическими соглашениями. Если противоречий нет, то тройка добавляется в список, представляющий собой расписание пассажирских авиаперевозок.

### Сценарий диалога с решателем задач

На рисунке 15 представлен возможный интерфейс решателя задач.

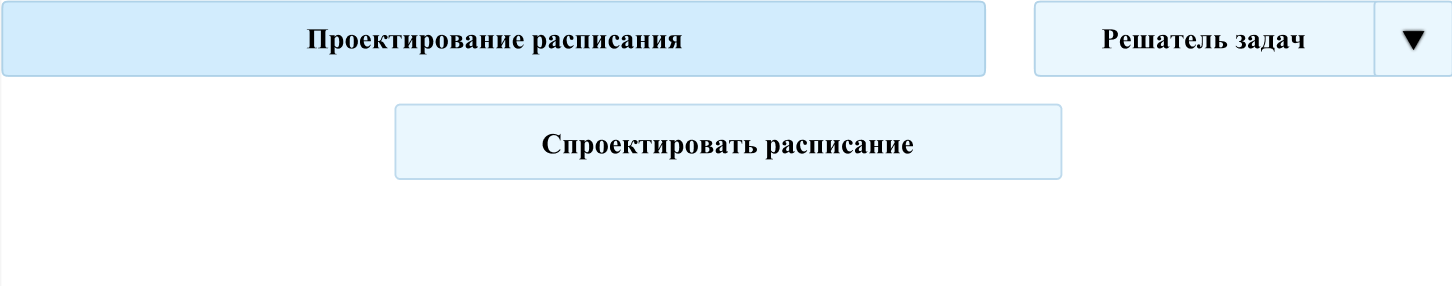


Рисунок 15 – Проект интерфейса решателя задач

Решатель задач, представленный на рисунке 15, позволяет составить расписание пассажирских авиаперевозок на основании введенных данных. При нажатии на кнопку «Спроектировать расписание» решатель задач обрабатывает знания и исходные данные согласно приведенному выше описанию. В результате работы решателя задач может получиться больше одного расписания пассажирских авиаперевозок, поэтому пользователь может их просмотреть, нажав на кнопку «Показать», и выбрать тот, что больше ему подходит, а затем нажать на кнопку «Сохранить» (см. Рисунок 16). После этого будет сформирован файл «Расписание\_пассажирских\_авиаперевозок.xlsx».

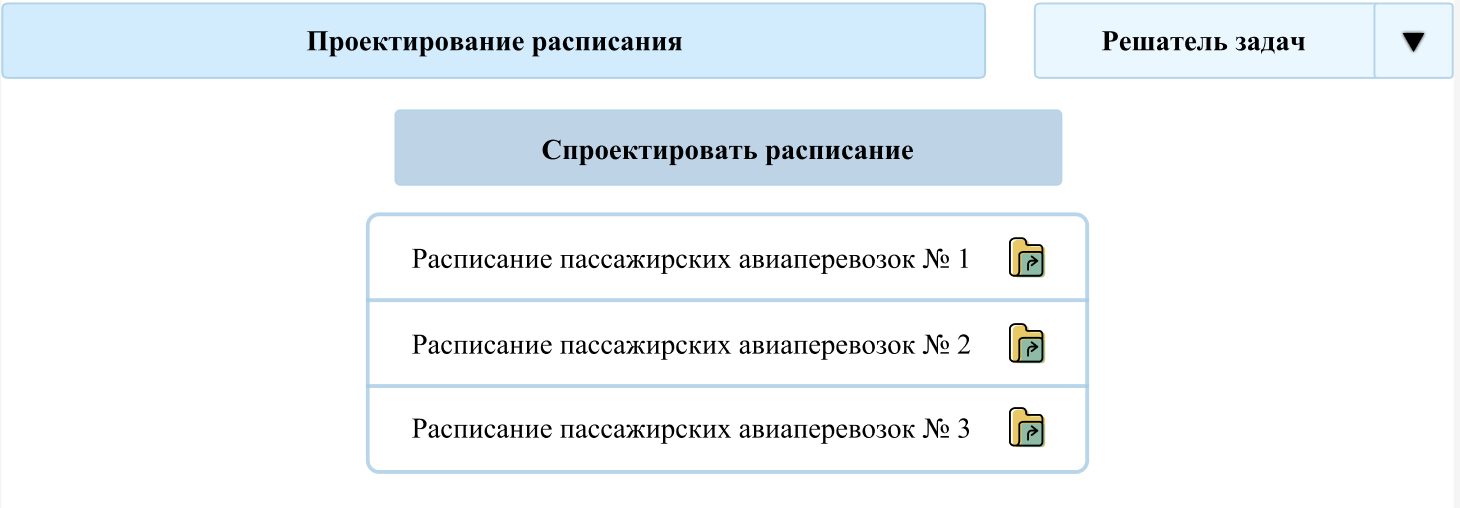


Рисунок 16 – Интерфейс выбора расписания пассажирских авиаперевозок

# Реализация и тестирование

## Средства разработки

Для реализации данного программного продукта была выбрана среда разработки *Microsoft Visual Studio Code*.

*Microsoft Visual Studio Code (VS Code)* – бесплатный текстовый редактор, разработанный *Microsoft* для операционных систем *Windows*, *Linux* и *macOS*. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с *Git*, подсветку синтаксиса, автодополнение кода, быструю навигацию по проекту, интегрированную консоль и многое другое.

Проект реализован на языке программирования *Python* – это высокоуровневый язык программирования, отличающийся эффективностью, простотой и универсальностью использования. Он широко применяется в разработке веб-приложений и прикладного программного обеспечения, а также в машинном обучении и обработке больших данных.

При реализации проекта также были использованы следующие технологии:

1. Flask – фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2;

2. Pickle – это модуль языка программирования Python, используемый для сериализации (преобразования объекта в поток байтов) и десериализации (преобразования потока байтов в объект) объектов Python. Pickle позволяет сохранять состояния объектов Python в файлы и загружать их обратно в память.

## Тестирование

Находясь в режиме «Редактор знаний» пользователь может вводить определенные данные. Если пользователь вводит данные, которые уже есть базе данных или, наоборот, которых нет, то система должна выдавать ошибку, когда пользователь будет пытаться спроектировать расписание пассажирских авиаперевозок. Если пользователь не допустил ошибки, то система должна спроектировать расписание пассажирских авиаперевозок.

Пользователь ввел город, который уже есть в базе городов (см. Рисунок 17). Следовательно, при его попытке спроектировать расписание система должна выдать ошибку (см. Рисунок 18). Аналогично с удалением города. Необходимо добавлять города, которых нет в базе городов.

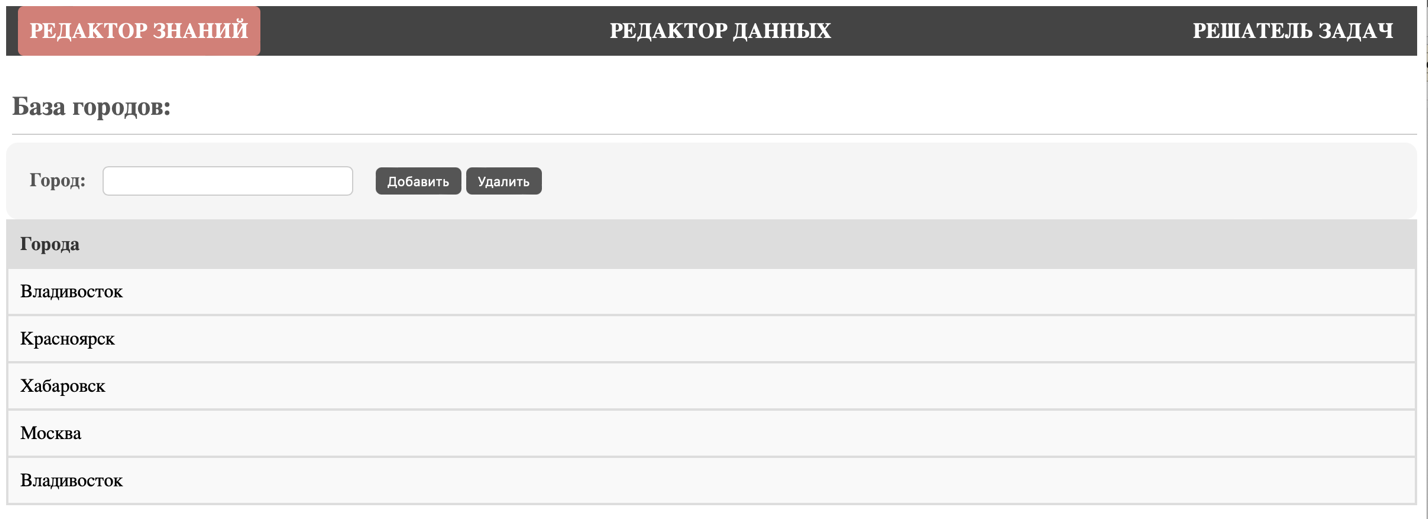


Рисунок 17 – Добавление города, который уже присутствует в базе городов

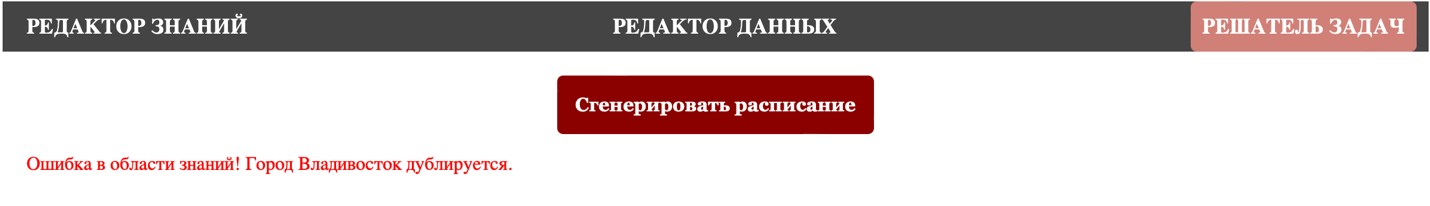


Рисунок 18 – Ошибка при попытке добавления уже имеющегося в базе города

Пользователь пытается удалить город, у которого имеются маршруты (см. Рисунок 19). Следовательно, при попытке спроектировать расписание система должна выдать ошибку (см. Рисунок 20). Для того, чтобы удалить город, необходимо сначала удалить все его маршруты, а после удалить город из базы городов.

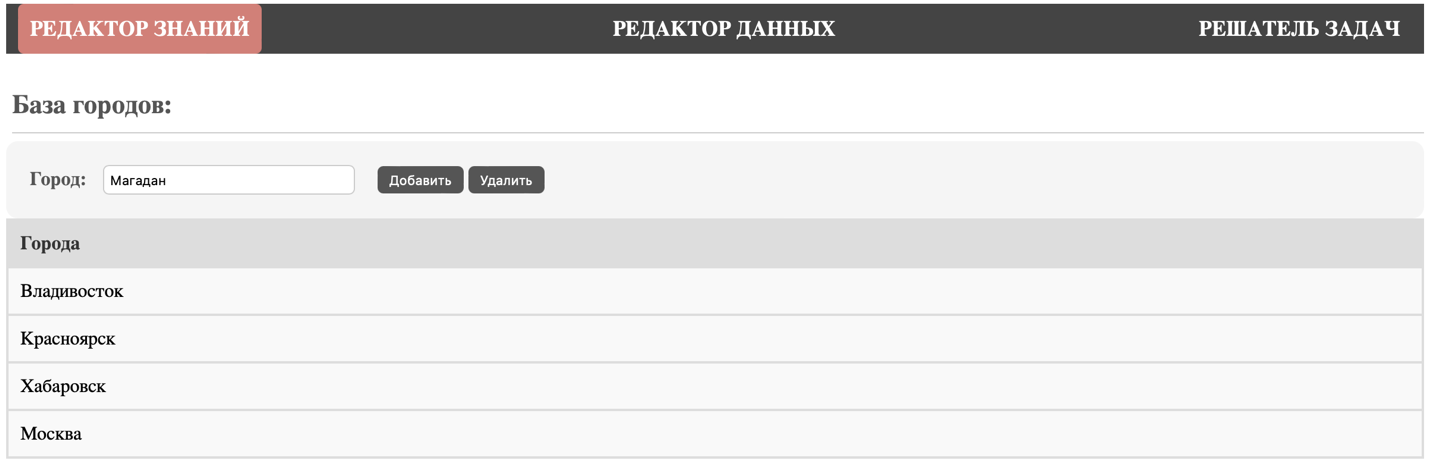


Рисунок 19 – Удаление города с имеющимися у него маршрутами

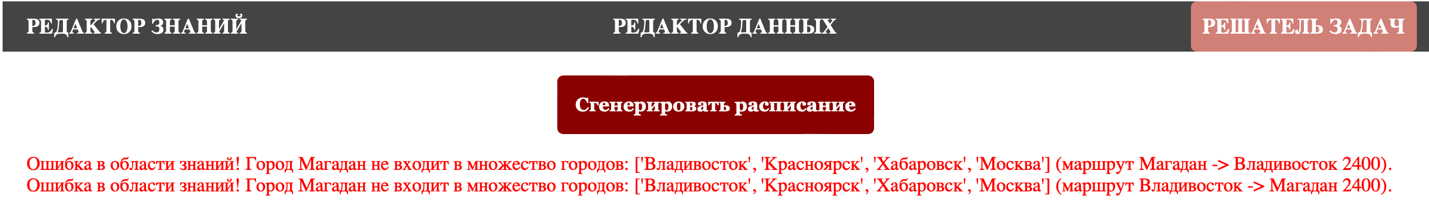


Рисунок 20 – Ошибка при попытке удаления города с имеющимися у него маршрутами

Пользователь хочет добавить маршрут, где отсутствует в базе городов город, который он указывает (см. Рисунок 21). Следовательно, при попытке спроектировать расписание система должна выдать ошибку (см. Рисунок 22). Аналогично с удалением маршрута. Необходимо указывать только те города, которые есть в базе городов.

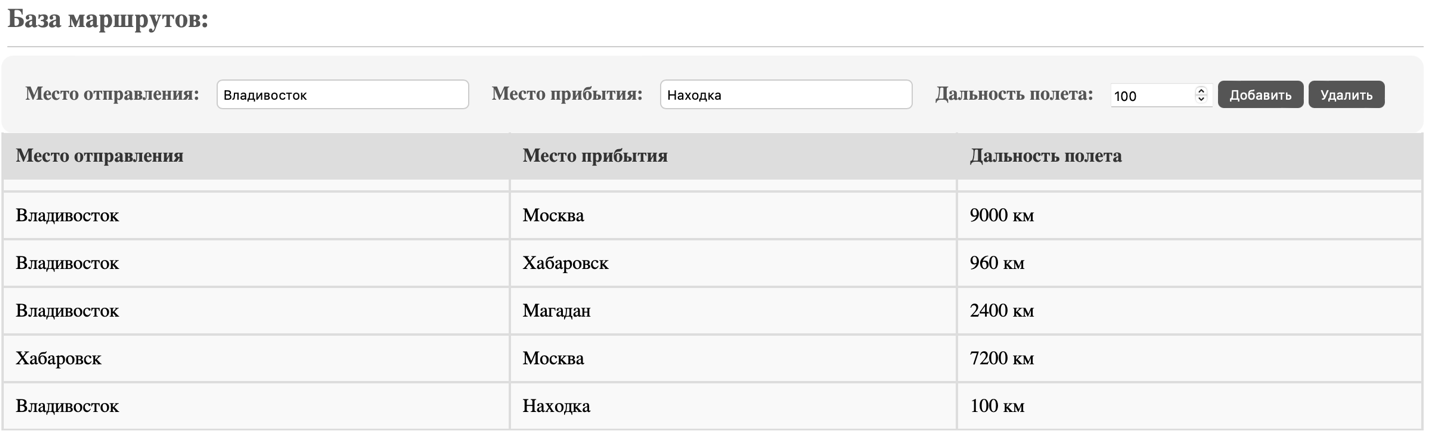


Рисунок 21 – Добавление маршрута с указанием города, которого нет в базе городов

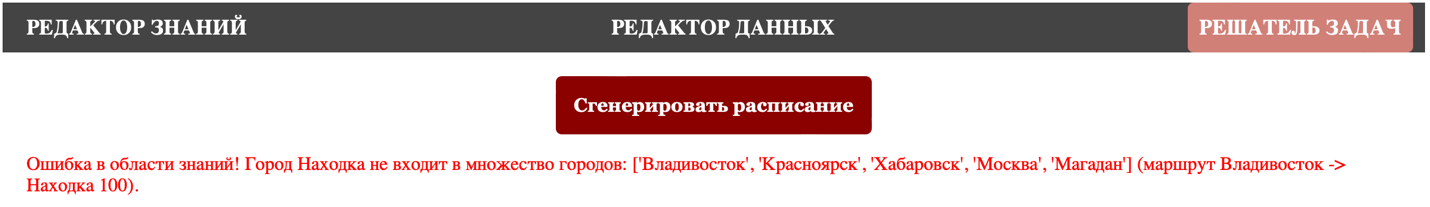


Рисунок 22 – Ошибка при попытке добавления маршрута с указанием города, которого нет в базе городов

Пользователь пытается добавить маршрут, уже имеющийся в базе маршрутов (см. Рисунок 23). Следовательно, при попытке спроектировать расписание система должна выдать ошибку (см. Рисунок 24). Необходимо добавлять маршруты, которых нет в базе маршрутов.

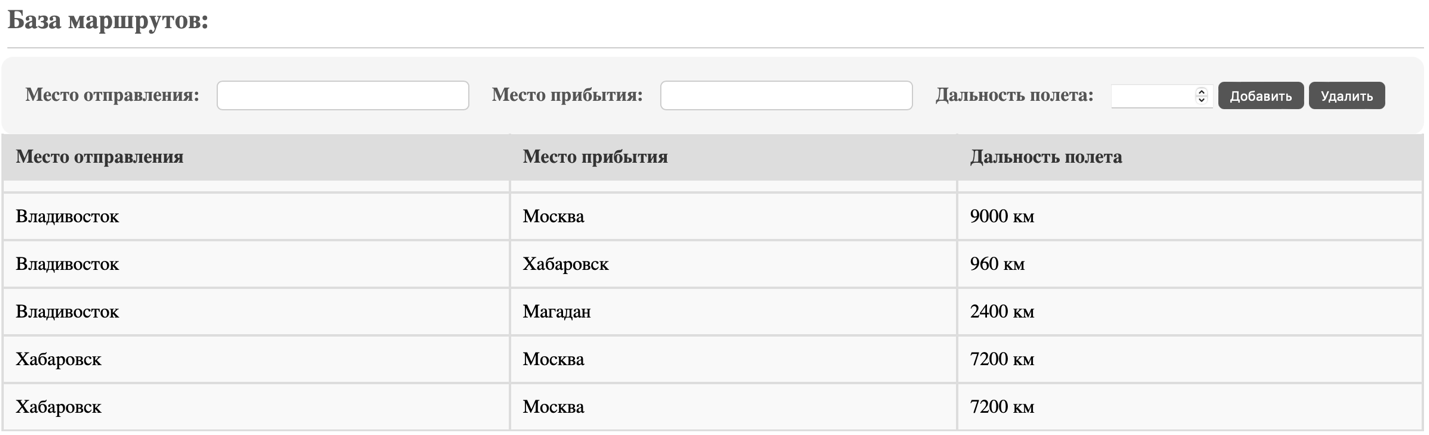


Рисунок 23 – Добавление маршрута, который уже присутствует в базе маршрутов

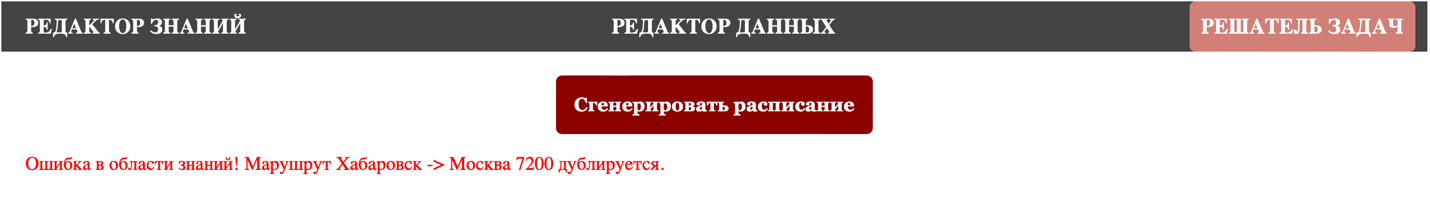


Рисунок 24 – Ошибка при попытке добавления маршрута, который уже присутствует в базе маршрутов

Пользователь пытается добавить тип самолета, уже имеющийся в базе типов самолета (см. Рисунок 25). Следовательно, при попытке спроектировать расписание система должна выдать ошибку (см. Рисунок 26). Аналогично с удалением типа самолета. Необходимо добавлять типы самолета, которых нет в базе маршрутов.



Рисунок 25 – Добавление типа самолета, который уже присутствует в базе типов самолета

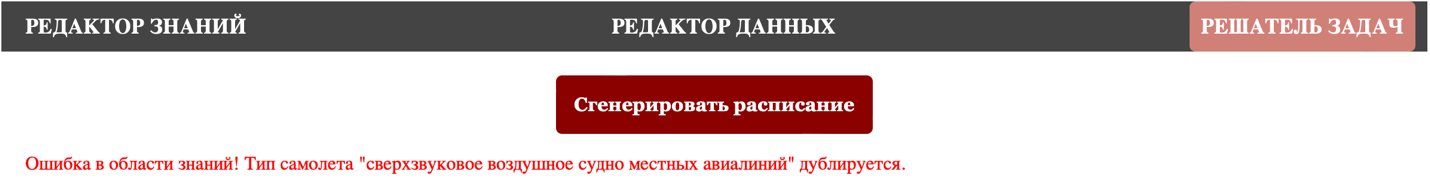


Рисунок 26 – Ошибка при попытке добавления типа самолета, который уже присутствует в базе типов самолета

Если пользователь не допустил никаких ошибок в редакторе знаний, то тогда будет спроектировано расписание пассажирских авиаперевозок (см. Рисунок 27).

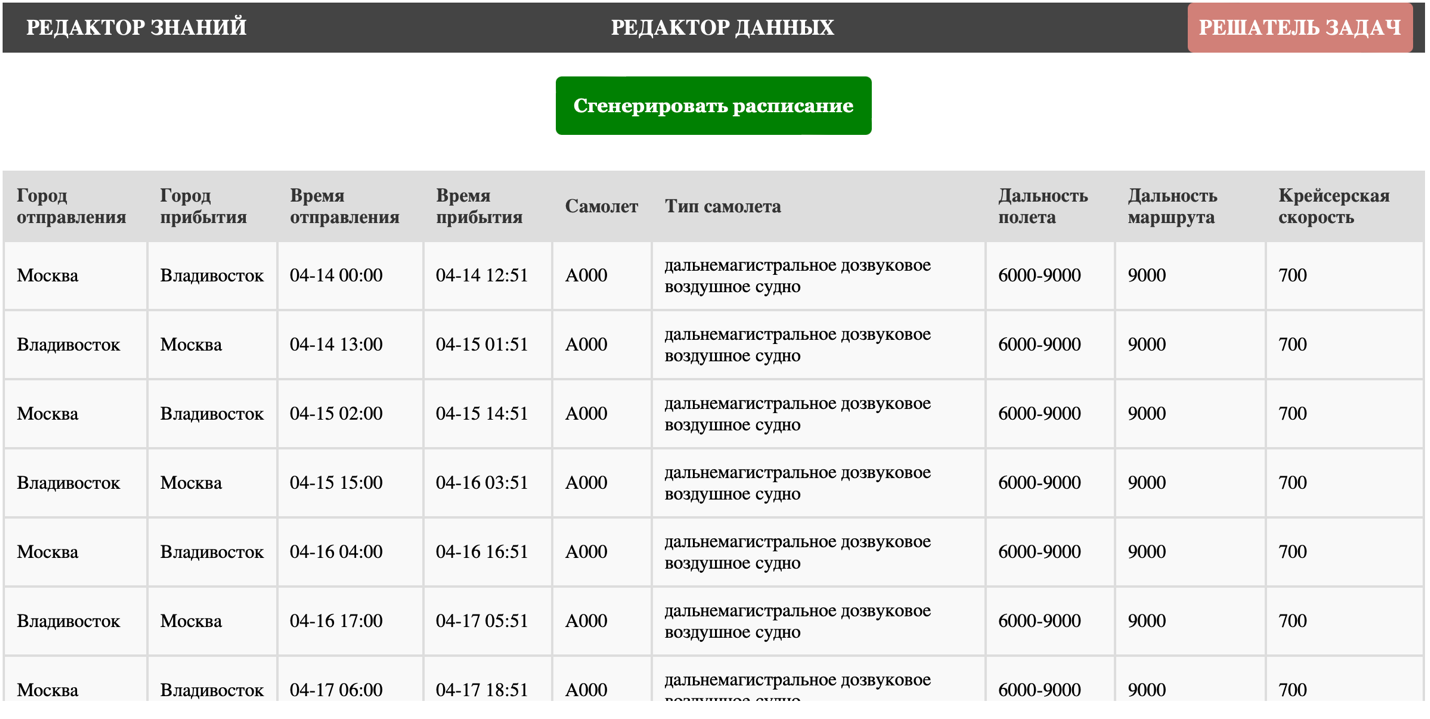


Рисунок 27 – Спроектированное расписание пассажирских авиаперевозок

# Заключение

В рамках лабораторной работы был разработан проект системы, основанной на знаниях, для чего были решены поставленные задачи:

* + 1. проанализирована предметная область;
    2. разработана модель предметной области;
    3. разработан проект системы и пользовательского интерфейса;
    4. реализовано программное средство и проведено тестирование.

Таким образом, цель данной лабораторной работы была достигнута.

# Список литературы

1. Артемьева И.Л. «Методы системного анализа и моделирования. Методические указания по выполнению самостоятельной работы и индивидуальных заданий. – 2019.».
2. Артемьева И.Л., Высоцкий В.Н., Рештаненко Н.В. «Модель онтологии предметной области (на примере органической химии)».
3. Правила перевозки пассажиров на воздушных линиях [Электронный ресурс]. // Режим доступа: https://aviacenter.ru/files/docs/pravila\_avia.pdf.