Федеральное государственное бюджетное образовательное учреждение

высшего образования (ФГБОУ ВО)

«Дальневосточный государственный университет путей сообщения»

Кафедра: «Информационные технологии и

системы»

Лабораторная работа № 3

Тема: «Разработка диаграммы классов»

Выполнил: Волков А.Э.

гр.СО251КОБ

Проверил: Анисимов В.В.

Сазанова Е.В.

Хабаровск, 2020

**Задание на выполнение лабораторной работы**

Разработать диаграммы классов по индивидуальному заданию:

- диаграмму классов анализа, описывающую информационную систему в целом;

- логическую (по-русски) и физическую (с учетом выбранной СУБД) диаграммы классов, описывающие структуру БД;

- логическую (по-русски) и физическую (с учетом выбранного языка программирования) диаграммы классов, описывающие структуру клиентского ПО.

**Ход выполнения работы**

***Диаграмма классов анализа***

Класс анализа – это укрупненная абстракция, которая на концептуальном уровне (без точного определения атрибутов и операций) описывает некоторый фрагмент системы.

Существует три вида классов анализа:

* граничный;
* управляющий;
* сущности.

Диаграмма классов анализа является прообразом классической диаграммы классов. Элементами, отображаемыми на диаграмме, являются классы и отношения между ними.

Назначение классов анализа:

* граничный класс – используется для моделирования взаимодействия между системой и актерами (пользователями, внешними системами или устройствами);
* управляющий класс – отвечает за координацию, взаимодействие и управление другими объектами, выполняет сложные вычисления, управляет безопасностью, транзакциями и т. п.;
* класс сущности – используется для моделирования долгоживущей, нередко сохраняемой информации. Классы сущности являются абстракциями основных понятий предметной области – людей, объектов, документов и т. д., как правило, хранимых в табличном или ином виде.

Связи между классами анализа отображаются с использованием отношений пяти видов:

* ассоциаций – показывает, что объекты одного класса содержат информацию о существовании (наличии в памяти) объектов другого класса и между ними имеется некоторая логическая или семантическая связь;
* агрегаций – указывает на отношение «часть–целое» и отображается сплошной линией с не закрашенным ромбиком со стороны «целого»;
* композиций – аналогично агрегации, в которой «части» не могут существовать отдельно от «целого»;
* обобщения – является обычным таксонометрическим отношением между более общим (абстрактным) классом (родителем или предком) и его частным случаем (дочерним классом или потомком);
* зависимостей – означает, что в спецификации или теле методов объектов одного класса (зависимого) выполняется обращение к атрибутам, методам или непосредственно к объектам другого класса.

Диаграмма классов анализа изображена на рисунке 1.

На диаграмме в виде классов отображено клиент-серверное взаимодействие между клиентским приложением и сервером. Соединение обеспечивает некоторый управляющий класс, который позволяет всем граничным классам приложения получать информацию, которая содержится на сервере, и наоборот, отправлять информацию на сервер для дальнейшей ее обработки. Все граничные классы представляют собой React-компоненты, которые отображаются на странице. В свою очередь все сущности представляют собой таблицы в базе данных.

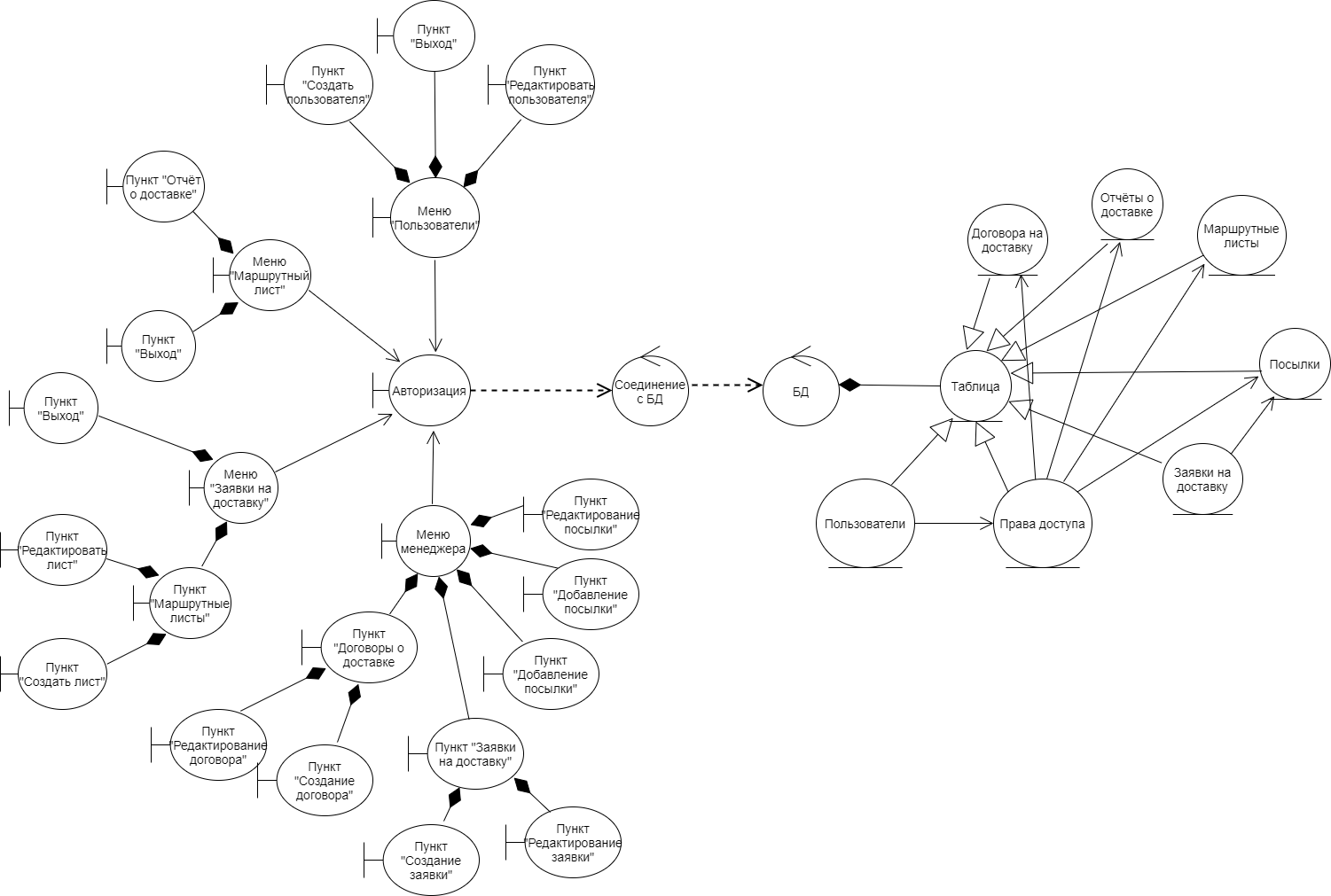


Рисунок 1. Диаграмма классов анализа

**Диаграммы классов**

Диаграммы классов используются при моделировании информационных систем наиболее часто. Они являются одной из форм статического описания системы с точки зрения ее проектирования, показывая ее структуру. Диаграмма классов не отображает динамическое поведение объектов, изображенных на ней классов.

На диаграммах классов показываются классы, интерфейсы и отношения между ними.

Диаграмма классов представляет собой граф, вершинами которого являются элементы типа «классификатор», связанные различными типами структурных отношений.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

* концептуальная точка зрения – диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
* точка зрения спецификации – диаграмма классов применяется при проектировании информационных систем;
* точка зрения реализации – диаграмма классов содержит классы, используемые непосредственно в программном коде.

Классы могут иметь логическую и физическую реализации. Логические диаграммы классов в отличие от физических, строятся без привязки к языкам программирования.

В ходе выполнения курсового проекта было разработано два типа диаграмм классов: для клиентского приложения и для серверного приложения.

Прежде чем перейти к разработке этих двух моделей, опишем структуру таблиц базы данных:

Таблица 1. Users(пользователи)

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| ID пользователя | int | Уникальный идентификатор пользователя |
| Фамилия | nvarchar | Фамилия пользователя |
| Имя | nvarchar | Имя пользователя |
| Отчество | nvarchar | Отчество пользователя |
| Менеджер | boolean | Является ли пользователь менеджером |
| Логист | boolean | Является ли пользователь логистом |
| Администратор | boolean | Является ли пользователь администратором |
| Курьер | boolean | Является ли пользователь курьером |
| Логин | nvarchar | Логин пользователя |
| Пароль | nvarchar | Пароль пользователя |
| Должность | nvarchar | Должность пользователя |

Таблица 2. Отчёты о доставке(Dilivery reports)

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| ID отчёта | int | Уникальный идентификатор отчёта |
| ID заявки | int | Ссылка на заявку |
| Дата доставки | Data Time | Дата доставки |

Таблица 3. Договора на доставку(Delivery contract)

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| ID contract | int | Уникальный идентификатор контракта |
| Фамилия отпр | nvarchar | Фамилия отправителя |
| Имя отпр | nvarchar | Имя отправителя |
| Отчество отпр | nvarchar | Отчество отправителя |
| Телефон отпр | nvarchar | Телефон отправителя |
| Адрес доставки | nvarchar | Адрес доставки |

Таблица 4. Маршрутные листы(Route maps)

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| ID маршрутного листа | int | Уникальный идентификатор листа |
| ID пользователя | int | Ссылка на пользователя |
| Статус | Boolean | Обработана ли заявка |

Таблица 5. Заявки на доставку(Delivery orders)

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| ID заявки | int | Уникальный идентификатор заявки |
| ID маршрутного листа | int | Ссылка на маршрутный лист |
| Дата создания | Data Time | Дата создания заявки |
| Срочность | boolean | Нужно ли выполнять заявку в первой |

Таблица 6. Посылки(Parcels)

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| ID послыки | int | Уникальный идентификатор посылки |
| ID заявки | nvarchar | Ссылка на заявку |
| ID договора | nvarchar | Ссылка на договор |
| Вес | int | Вес посылки |
| Высота | int | Высота посылки |
| Ширины | int | Ширина посылки |
| Длина | int | Длина посылки |
| Хрупкая | nvarchar | Хрупкая ли посылка |
| Фамилия получателя | nvarchar | Фамилия получателя |
| Имя получателя | nvarchar | Имя получателя |
| Отчество получателя | nvarchar | Отчество получателя |
| Телефон получателя | nvarchar | Телефон получателя |

Далее приведены диаграммы классов для БД (рис. 4 и рис. 5) и диаграммы классов для приложения (рис. 2 и рисунок 3). Каждый вид диаграмм представлен в двух экземплярах: логическом (на русском языке) и физическом (с учетом языка программирования).

Целевой СУБД для описания диаграмм классов БД, является PostgreSQL.

Целевым языком программирования для описания диаграмм классов приложения является Python (Django).

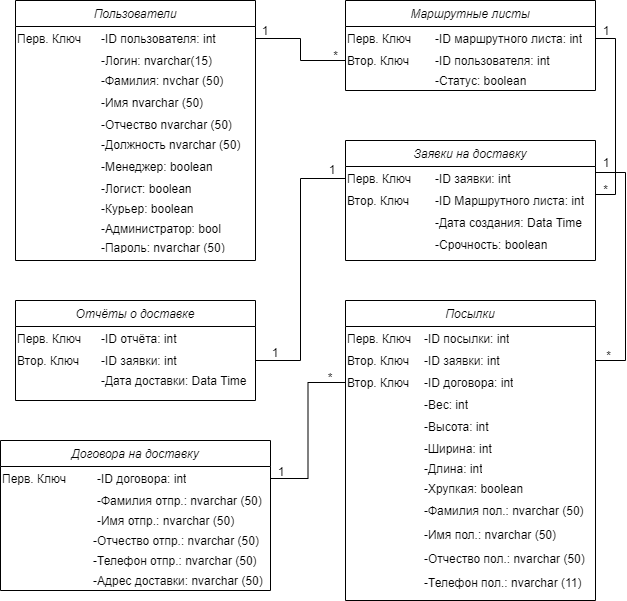


Рисунок 2. Логическая диаграмма классов приложения

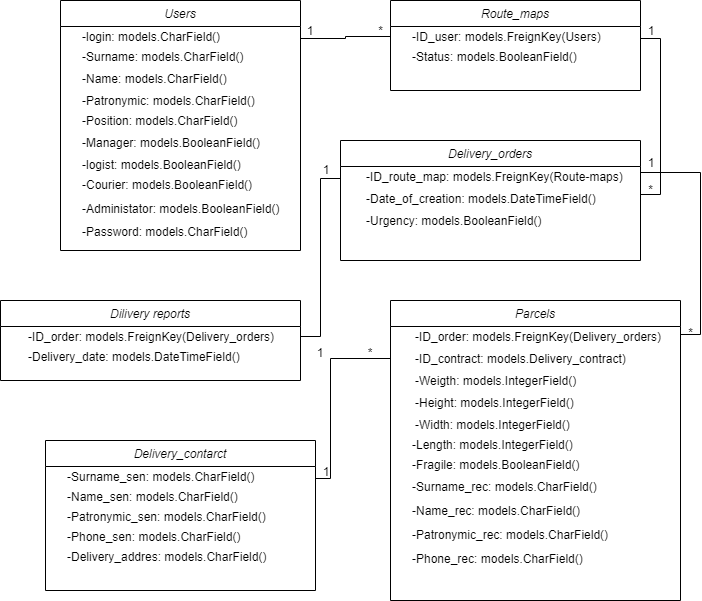


Рисунок 3. Физическая диаграмма классов приложения

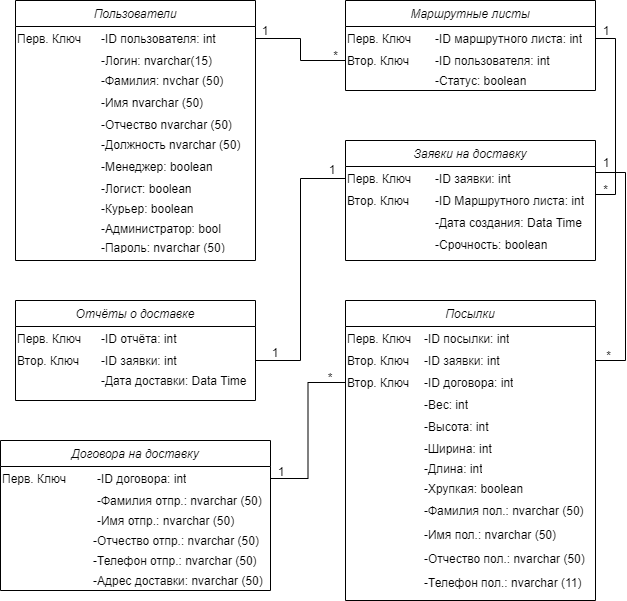


Рисунок 4. Логическая диаграмма классов базы данных

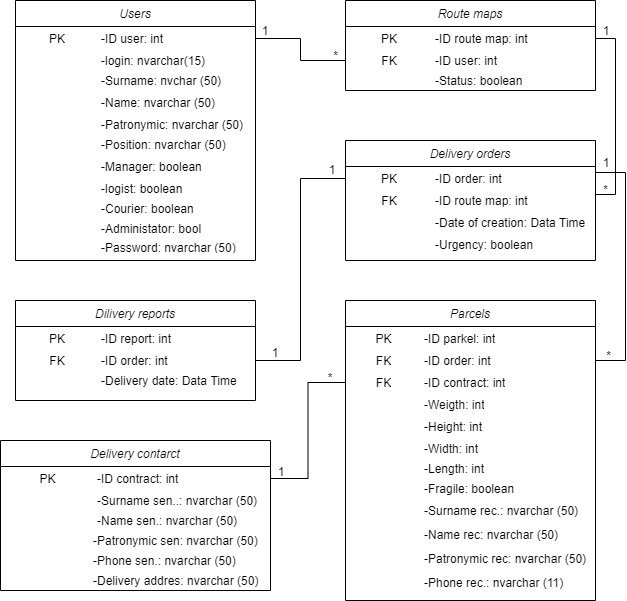


Рисунок 5. Физическая диаграмма классов базы данных