**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Сибирский государственный университет науки и технологий**

**имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

институт/ факультет/ подразделение

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

по дисциплине: «Управление жизненным циклом программных систем»

Тема: «Разработать проект системы в UML»

Преподаватель \_\_\_\_Н.С. Черниченко\_

подпись, дата инициалы, фамилия

Обучающийся МПА23-01 \_\_\_\_А.А. Першин\_

номер группы подпись, дата инициалы, фамилия

Красноярск 2023

**Цель работы:** сформировать навыки разработки проекта информационных систем в UML по теме «Разработка автоматизированной системы управления транспортными услугами».

**Задачи:**

1. Спроектировать функциональные требования к разрабатываемой системе в виде модели Use Case;
2. Спроектировать модель Activity diagram;
3. Спроектировать модель Sequence diagram;
4. Спроектировать Class diagram;
5. Спроектировать модель Deployment diagram;
6. Спроектировать модель размещения.

**Ход работы:**

**Диаграмма прецедентов** (Use Case) использует 2 основных элемента:

1) Actor (участник) — множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Участником может быть человек, роль человека в системе или другая система, подсистема или класс, которые представляют нечто вне сущности.

2) Use case (прецедент) — описание отдельного аспекта поведения системы с точки зрения пользователя. Прецедент не показывает, "как" достигается некоторый результат, а только "что" именно выполняется.

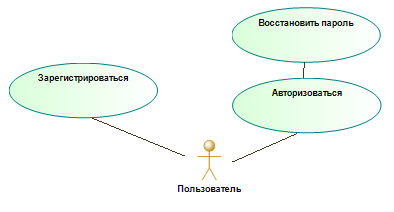
****

Рисунок 1 – Регистрация. Use Case

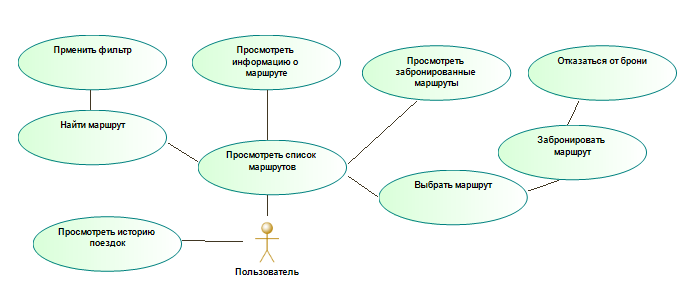


Рисунок 2 – Покупка билета. Use Case



Рисунок 3 – Администратор. Use Case

**Диаграмма активностей** описывает динамические аспекты поведения системы в виде блок-схемы, которая отражает бизнес-процессы, логику процедур и потоки работ — переходы от одной деятельности к другой. По сути, мы рисуем алгоритм действий (логику поведения) системы или взаимодействия нескольких систем. Ниже —диаграмма разрабатываемой системы.

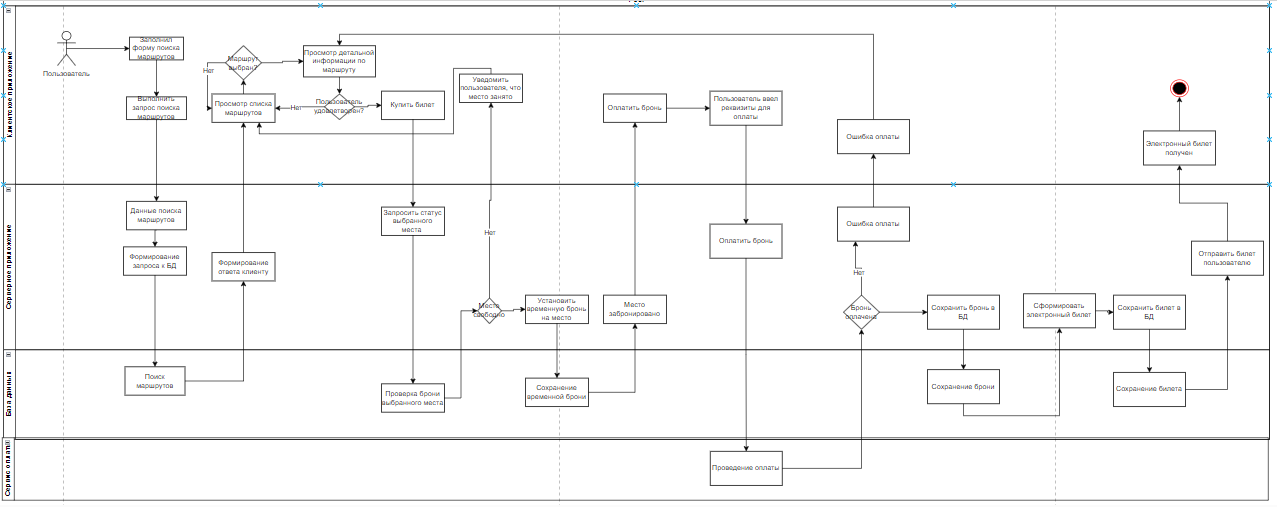


Рисунок 4 – Диаграмма активности

Подробно диаграмму можно рассмотреть по ссылке <https://drive.google.com/file/d/1PSxfcCYfb71OOat6NcRUdMhZg8T0td23/view?usp=sharing>

**Диаграмма последовательности** используется для уточнения диаграмм прецедентов — описывает поведенческие аспекты системы. Диаграмма последовательности отражает взаимодействие объектов в динамике, во времени. При этом информация принимает вид сообщений, а взаимодействие объектов подразумевает обмен этими сообщениями в рамках сценария.

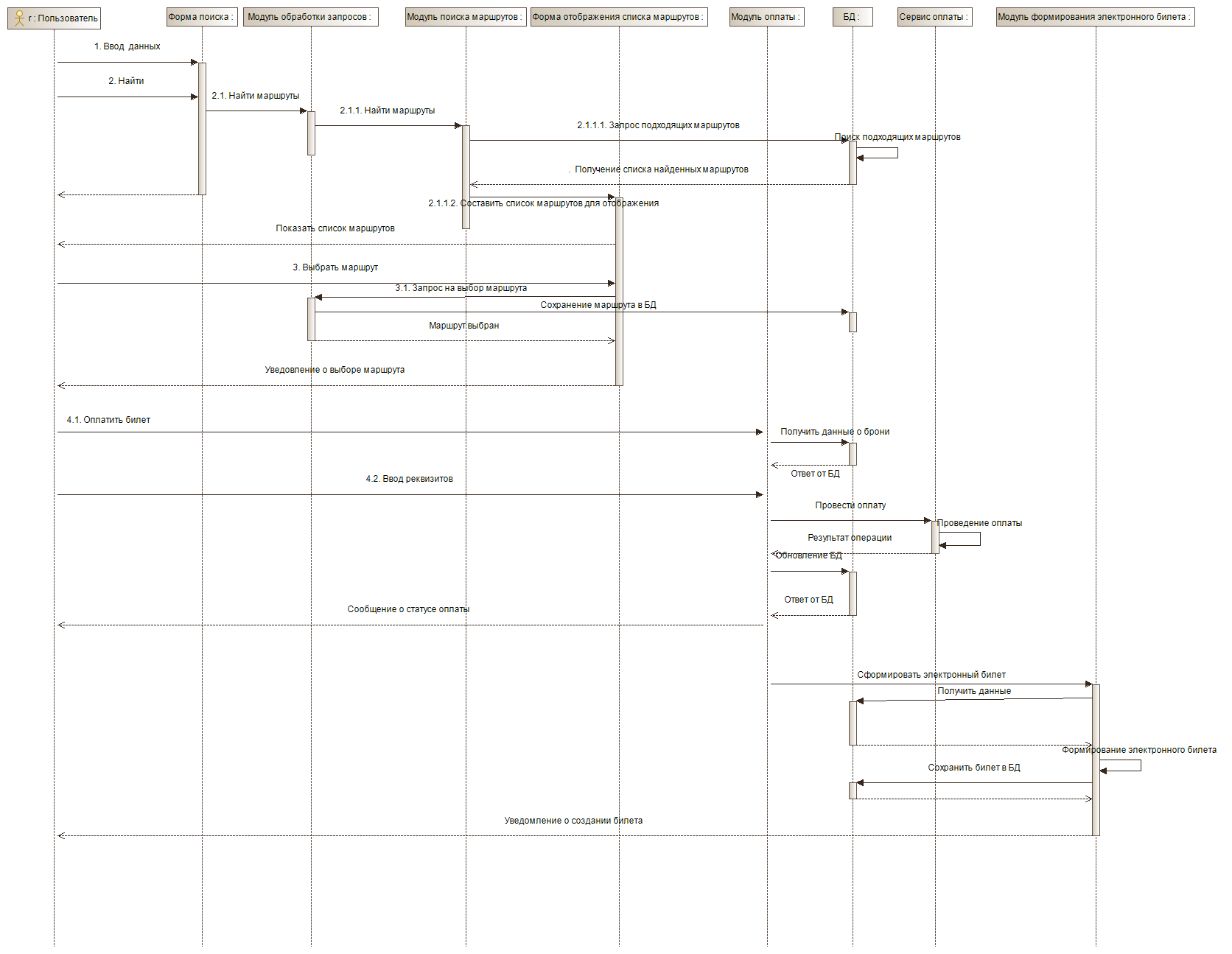


Рисунок 5 – Диаграмма последовательности

**Класс (class)** — категория вещей, которые имеют общие атрибуты и операции. Сама диаграмма классов являет собой набор статических, декларативных элементов модели. Она дает нам наиболее полное и развернутое представление о связях в программном коде, функциональности и информации об отдельных классах. Приложения генерируются зачастую именно с диаграммы классов.

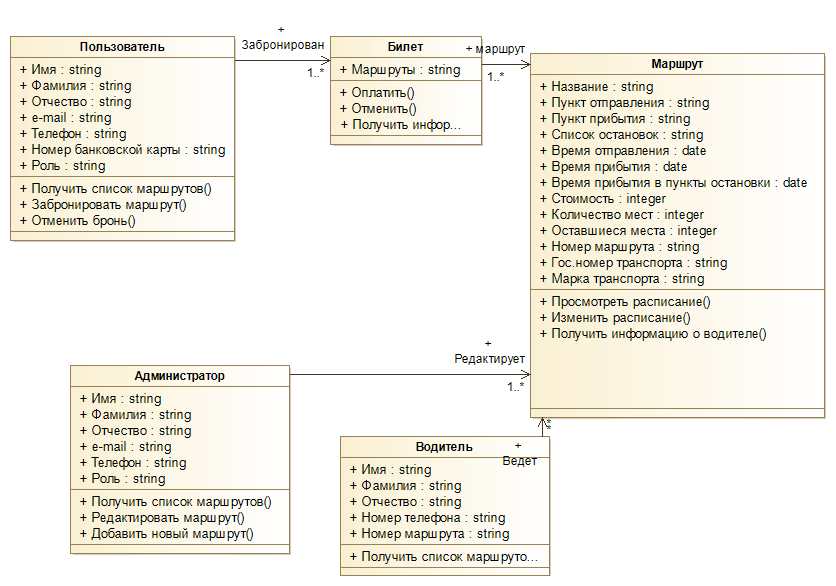


Рисунок 6 – Диаграмма классов

**Диаграмма развертывания** отображает графическое представление инфраструктуры, на которую будет развернуто приложение: топологию системы и распределение компонентов по ее узлам, а также соединения — маршруты передачи данных между узлами. Диаграмма помогает более рационально организовать компоненты, от чего зависит в числе прочего и производительность системы, а также решить вспомогательные задачи, например, связанные с безопасностью.

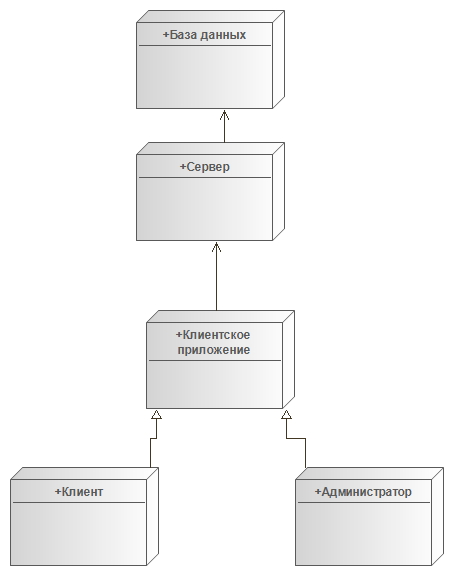


Рисунок 7 – Диаграмма развертывания

В UML **диаграммы размещения** используются для визуализации статических аспектов физических узлов и их взаимосвязей, а также для описания их деталей, которые имеют отношения к конструированию систем.

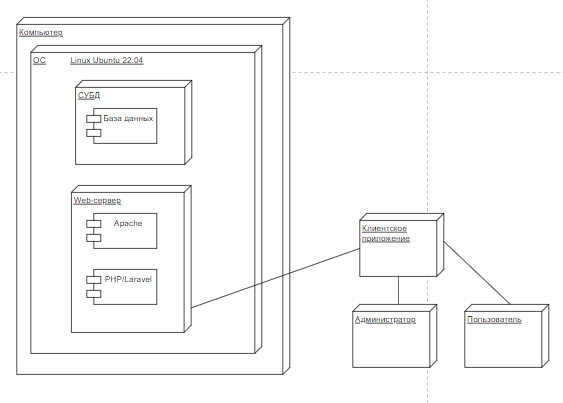


Рисунок 8 – Диаграмма размещения

**Вывод**

В результате выполненной лабораторной работы были получены знания и навыки разработки проекта информационной системы на языке UML. Были спроектированы диаграммы для описания «внутренности» системы.