Министерство образования и науки Российской Федерации федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики» (Университет ИТМО)

Факультет: Инфокоммуникационных технологий **Направление (специальность)** 11.04.02 «Инфокоммуникационные технологии и системы связи»

ДОМАШНЕЕ ЗАДАНИЕ

По дисциплине: «Проектирование информационных систем»

На тему: «Разработка программного прототипа по проекту инфокоммуникационной системы»

Выполнил:

студент(ка) гр. K4113c Логинова В. С.

Преподаватель:

Осипов Н. А.

Санкт-Петербург 2020 **Цель работы:** Разработать уточненную (после проведения практических занятий и лабораторных работ) диаграмму классов.

Задачи:

Применить шаблоны GoF в тех случаях, где это полезно. Но для обучения и оценивания, чем больше, тем лучше (минимум должно быть реализовано 4 шаблона).

В отчете указать обоснование применения этих шаблонов.

Программное обеспечение: draw.io

Ход работы:

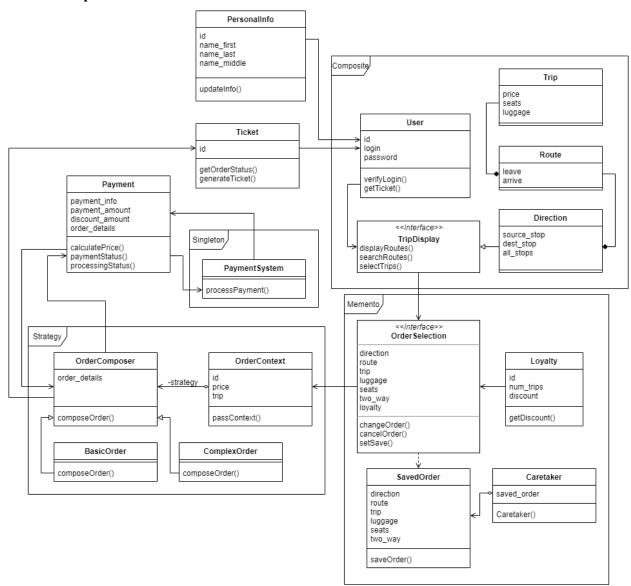


Рисунок 1 – Реализованная диаграмма классов

Паттерн **Composite** — Компонует объекты в древовидные структуры для представления иерархий «часть — целое». Позволяет клиентам единообразно трактовать индивидуальные и составные объекты. Реализацию паттерна можно представить в виде меню, которое имеет различные пункты. Эти пункты могут содержать подменю, в которых, в свою очередь, также имеются пункты. То есть пункт меню служит с одной стороны частью

меню, а с другой стороны еще одним меню. В итоге мы однообразно можем работать как с пунктом меню, так и со всем меню в целом.

Паттерн **Strategy** – определяет семейство алгоритмов, инкапсулирует каждый из них и делает их взаимозаменяемыми. Стратегия позволяет изменять алгоритмы независимо от клиентов, которые ими пользуются. Используется когда необходимо обеспечить выбор из нескольких вариантов решений, которые можно легко менять в зависимости от условий, и когда необходимо менять поведение классов и объектов на стадии выполнения программы.

Паттерн **Memento** — позволяет выносить внутреннее состояние объекта за его пределы для последующего возможного восстановления объекта без нарушения принципа инкапсуляции. Используется, когда нужно сохранить состояние объекта для возможного последующего восстановления.

Паттерн **Singleton** – через который можно получить доступ к стабильной справочной информации или некоторым утилитам. Гарантирует, что у класса есть только один экземпляр, и предоставляет точку доступа к нему.

Вывод:

В данной работе улучшили спроектированную диаграмму классов при помощи применения ряда изученных паттернов проектирования.