**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**ОТЧЁТ**

#### по домашнему заданию на тему: «Разработка программного прототипа по проекту инфокоммуникационной системы»

по дисциплине «Проектирование ИКС»

Выполнили: студенты группы к4113с  
Никитин Д.В

«23» сентября 2020 г. \_\_\_\_\_\_\_\_\_\_\_/Никитин Д.В./

Принял: Осипов Н.А.

«24» сентября 2020 г. \_\_\_\_\_\_\_\_\_\_\_/Осипов Н.А./

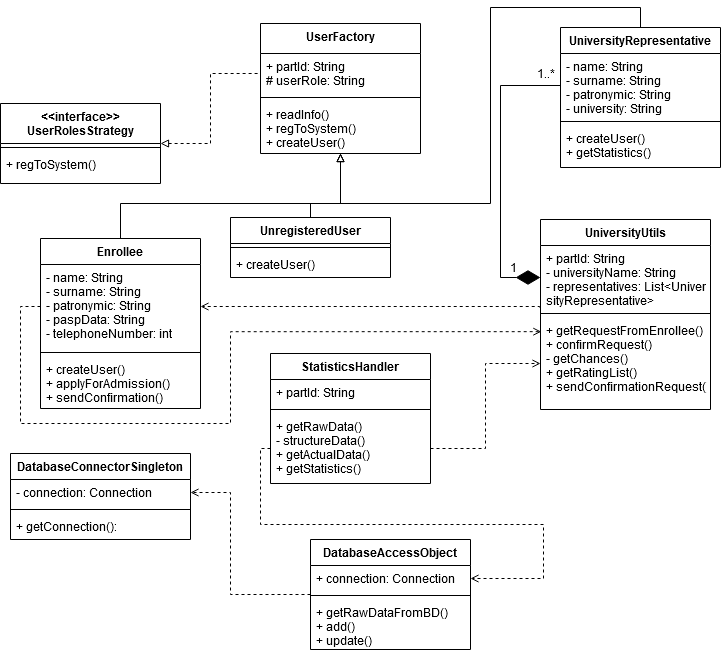
## **Задание на выполнение лабораторной работы**

Разработать уточненную (после проведения практических занятий и лабораторных работ) диаграмму классов. Реализовать классы как можно подробнее.

Применить шаблоны GoF в тех случаях, где это полезно.

**Вариант инфокоммуникационной системы**: единая государственная система абитуриента

1. Диаграмма классов



* **UserRolesStrategy -** интерфейс, отвечающий за выбор ролей
* **UserFactory -** задаёт своим подклассам такое поведение, чтобы этот метод выбора ролей реализовать. Имеет метод createUser()
* **UnregisteredUser -**  незарегистрированный пользователь, который ничего делать не может
* **Enrollee –** абитуриент, возможность подать заявку на поступление и отправить согласие на зачисление
* **UniversityRepresentative -** может только собрать статистику
* **UniversityUtils -** университетский класс, он отправляет согласия на зачисления, принимает заявки на поступление, формирует шансы и рейтинговые листы
* **StatisticHandler -** занимается чисто статистикой
* **DataBaseAccessObject –** объект доступа к базе данных
* **DatabaseConnector Singleton**

Используемые паттерны:

* **Стратегия -** позволяет изменять алгоритмы независимо от клиентов, которые ими пользуются. Данный паттерн следует применять, когда необходимо обеспечить выбор из нескольких вариантов решений, которые можно легко менять в зависимости от условий. В нашем случае выбор того, каким типом пользователя зарегистрироваться в системе
* **Фабричный метод** – применятся, когда создание новых объектов необходимо делегировать из базового класса классам наследникам. В нашем случае есть метод createUser() и с его помощью можно создавать пользователей с разным поведением (разными ролями). И этот метод переопределяется в разных видах пользователей
* **Синглтон** - обычно используется для обеспечения доступа к какому-либо ресурсу, который требуется разным частям приложения. В нашем случае данный паттерн используется для того, чтобы создать и удерживать в себе соединение с базой данных. От этого синглтона зависит объект доступа к БД, у которого как раз есть метод по получению данных из базы данных, и от него уже зависит StatisticsHandler