Министерство образования и науки

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Отчет

по лабораторной работе №4

по дисциплине **«Проектирование инфокоммуникационных сетей»**

Выполнил: **студент группы**

**K4113с Огирь К.С.**

Проверил: **Осипов Н.А.**

Санкт-Петербург

2020

# Лабораторная работа №4. Проектирование объектной модели

## Цель работы

Изучить основы разработки объектных моделей с использованием шаблонов GRASP для распределения обязанностей между классами.

## Ход работы

Для всех прецедентов, определенных в спецификации требований к программному обеспечению, описанному в лабораторной работе №3 необходимо было составить диаграмму последовательностей для представления системы как набора взаимодействующих объектов. Из-за большого количества прецедентов и многослойной архитектуры разрабатываемой системы модель была разделена и представлена на нескольких изображениях (рис. 1-3).

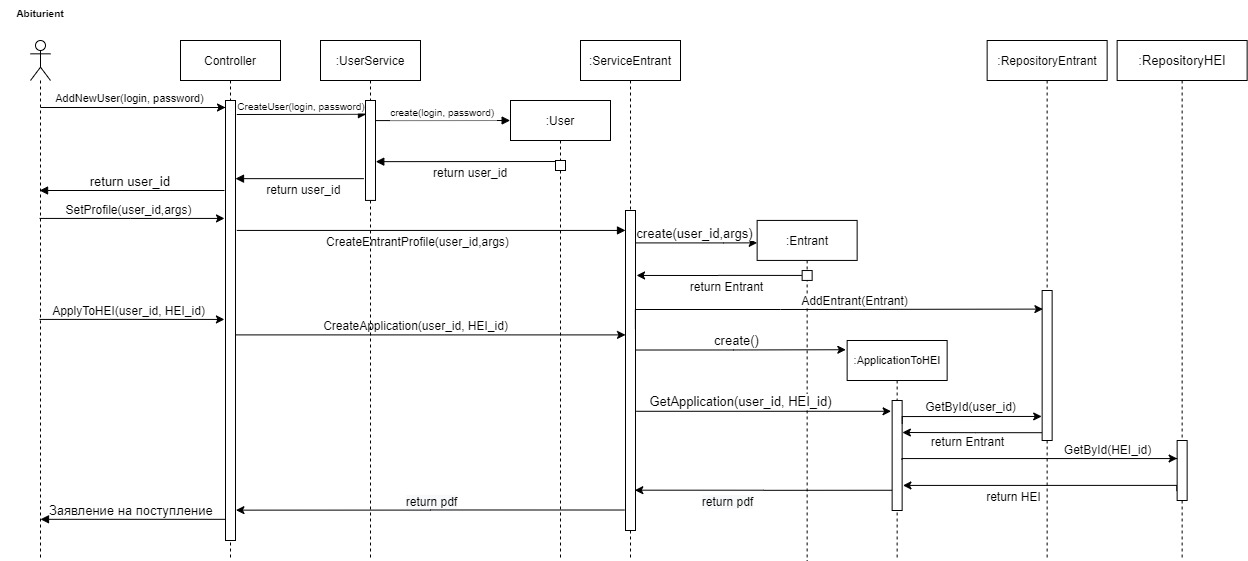


Рисунок – Диаграмма последовательностей для прецедентов абитуриента первая часть

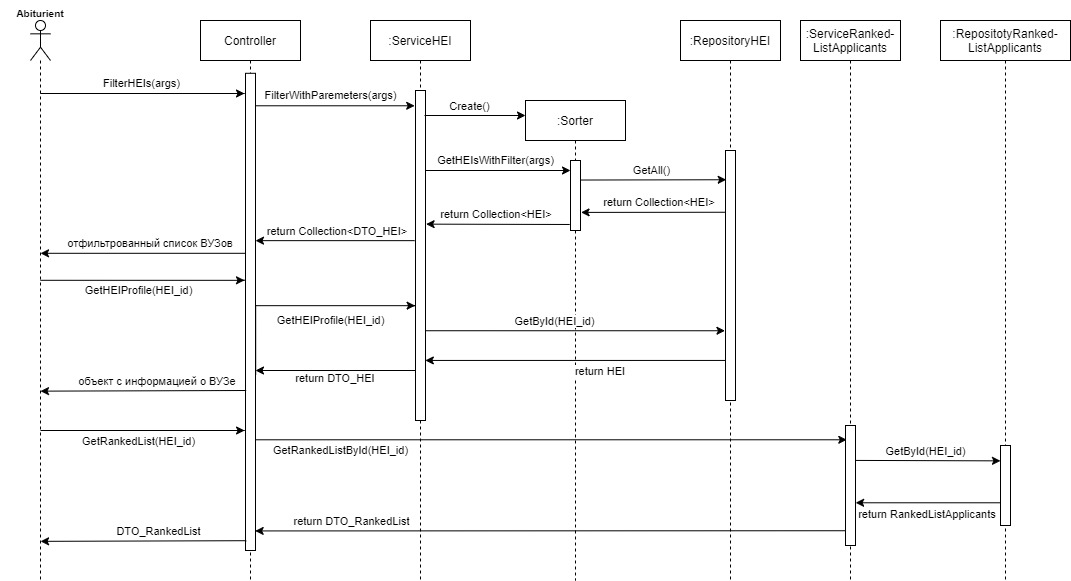


Рисунок – Диаграмма последовательностей для прецедентов абитуриента вторая часть

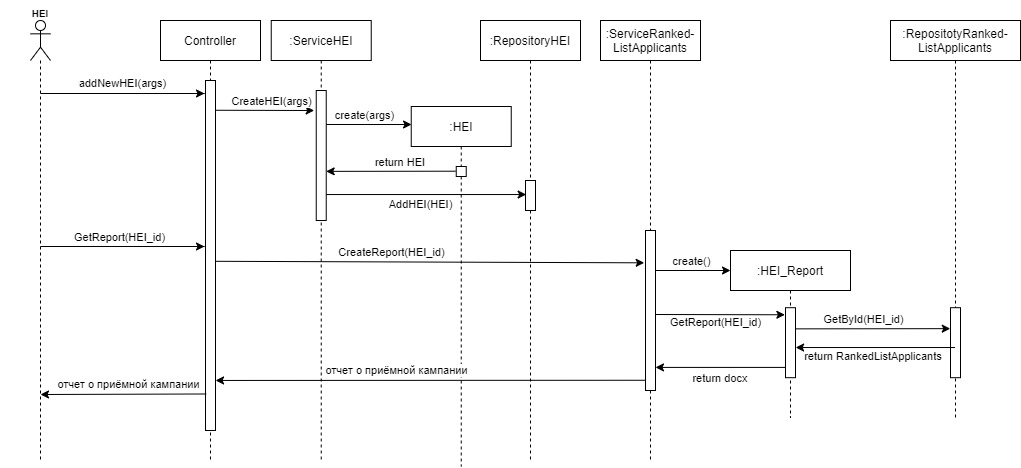


Рисунок – Диаграмма последовательностей для прецедентов ВУЗа

При создании диаграмм за основу была взято разделение системы на несколько слоёв: слой представления – слой, отвечающий за отображение переданной информации пользователю, а так-же получение информации от пользователя и отправка в нужном формате на контроллер; контроллер – слой, инкапсулирующий логику работы системы и её структуру от пользователя, он является внешним представлением всей системы для пользователя и не отвечает за обработку информации; модель – слой, отвечающий за работу с данными - ее обработку и хранение. Слой модель был разделен на несколько частей: часть, отвечающая за обработку информации – это слой бизнес-логики приложения (BLL), а также часть, отвечающая за запись и чтение информации из БД – слой доступа данных (DAL).

На диаграмме последовательностей контроллер везде представлен одним объектом, для достижения слабой связности и не создавать объект «всезнайку» контроллер был разделен на 2 независимых друг от друга класса.

Для понимания структуры приложения, какие объекты содержат или взаимодействуют с другими объектами, связей между классами была разработана модель диаграммы классов (приложение 1). Данная модель объясняет структуру и связи контроллера, BLL и DAL.

Классы, имеющие в названии слово «Service» - являются частью BLL, контроллеры системы взаимодействуют только со специальными сервисами, предоставляющими возможности системы для различных сущностей в системе. Данные классы также создают и используют объекты других разработанных классов для определенных специфических задач. Согласно шаблону GRASP такие классы можно отнести к типу «Creator».

Классы, имеющие в названии слово «Repository» - являются частью DAL, доступ к DAL имеет только BLL. Данные классы содержат в себе объекты данных из БД, согласно шаблону GRASP – данный слой воспринимается как «Information Expert», т. к. именно эти классы обладают данными.

Классы, содержащие в названии слово «Controller» - являются классами, воспринимаемыми согласно шаблону GRASP как «Controller».

Так же был применен применён шаблон «Low coupling», для примера рассмотрим ControllerAbiturient: он взаимодействует с сервисом ServiceHEI, и для получения каких-либо данных он обращается к нему, т. е. делегирует обязанность, не зная о том, что объект этого класса также не содержит данных, и сам обращается к информационному эксперту. Другой пример с этим же контроллером и сервисом: контроллер не подозревает, что при запросе на получение отфильтрованного списка абитуриентов объект сервиса создает и делегирует эту обязанность другому объекту. Таким образом при делегировании полномочий другим объектам достигается паттерн слабой связности.

Оптимальным при таком разделении задач и построении системы будет использование технологии .NET MVC, предоставляющей возможность легкого разделения системы на слои, а также предоставляющего возможность легкой реализации инверсии зависимостей через IoK контейнер.

## Вывод

В результате выполнения лабораторной работы были разобраны шаблоны GRASP и применены при создании UML-диаграммы последовательностей для прецедентов системы. В ходе моделирования система была разделена на несколько слоёв: слой контроллеров, слой бизнес-логики (BLL) и слой доступа к данным (DAL). В результате была получена точная модель, описывающая работу системы через взаимодействие объектов. На основе разработанной диаграммы была построена модель диаграммы классов с применением нотации UML, показывающая взаимосвязь между классами и описывающая все необходимые методы и свойства, необходимые для работы системы. На основе разработанной диаграммы должны быть реализованы классы и интерфейсы на языке C#, были окончательно определены используемые технологии при реализации: Entity Framework для работы с сущностями БД и позволяющий не задумываться о структуре и проектировании БД; .NET MVC для реализации слоёной структуры системы.

## Приложение 1

