



Министерство науки и высшего образования Российской  
Федерации Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический университет имени  
Н.Э. Баумана (национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

# **Лабораторная работа №5**

## **По курсу**

### **«Технологии разработки программного обеспечения»**

#### **«Разработка модели анализа»**

Подготовил:

Студент группы

**ИУ5-14Б Журавлев Н.В**

22.11.2023

Проверила:

**Виноградова М.В.**

2023 г.

## Цель работы:

- Изучить унифицированный процесс разработки (RUP);
- Приобрести умения построения модели анализа;
- Получить навыки построения модели анализа в среде Sparx Enterprise Architect.

## Полученное задание:

1. Открыть в среде Sparx Enterprise Architect проект, созданный ранее. Добавить к нему модель анализа RUP.
2. Добавить в модель анализа кооперации для реализации основных прецедентов. Для каждого прецедента одну кооперацию. Название кооперации совпадают с названием прецедента. Составить для основных прецедентов их описания (предусловия, поток событий, постусловие). Составить для основных прецедентов диаграммы деятельности (на основе описания).
3. Построить «обзорную» диаграмму коопераций (диаграмма классов).
4. Выполнить анализ архитектуры и построить в пакете Перспектива «обзорную» диаграмму классов сущностей (на основе модели предметной области). Классы сущностей копировать в модель из строительных блоков. Определить для них атрибуты, связи ассоциации, роли, множественность и арность.
5. Добавить в модель управляющие и граничные классы. Управляющий класс — один на каждую кооперацию. Граничный класс — один на каждого актера. Построить в пакете Перспектива «обзорную» диаграмму граничных классов и «обзорную» диаграмму управляющих классов.
6. Выполнить анализ коопераций. Для каждой кооперации: открыть диаграмму классов кооперации и переместить на нее классы (граничные, управляющий и сущностей), которые участвуют в этой кооперации. Определить ассоциации между классами в соответствии с их взаимодействием для реализации прецедента (общая схема: граничный управляющий сущности). Создать диаграмму последовательностей кооперации (для выполнения основного потока алгоритма). В качестве объектов использовать классы-участники кооперации и актера. (Первое действие всегда от актера.) События диаграммы будут прообразом его методов.
7. Проверить модель средствами пакета (валидация модели).
8. Выполнить анализ классов. После построения диаграммы последовательностей уточнить на диаграмме классов кооперации: связи

между классами, методы классов (названия и атрибуты), атрибуты классов.  
 Класс сущности: атрибуты поля хранимых данных, методы get|set или CRUD.  
 Класс граничный: атрибуты элементы формы (на основе эскиза пользовательского интерфейса)/параметры протокола, методы управляющие сигналами от пользователя (их обработчики). Классы управляющие: атрибуты промежуточные данные алгоритма, методы шаги алгоритма кооперации.

9. Проверить модель средствами пакета (валидация модели).
10. Построить «обзорную» диаграмму всех классов. Добавить в модель пакеты.  
 Распределить классы анализа по пакетам. Указать зависимости между пакетами.

## Ход работы:

### 1. Описание модели требований. Диаграмма прецедентов

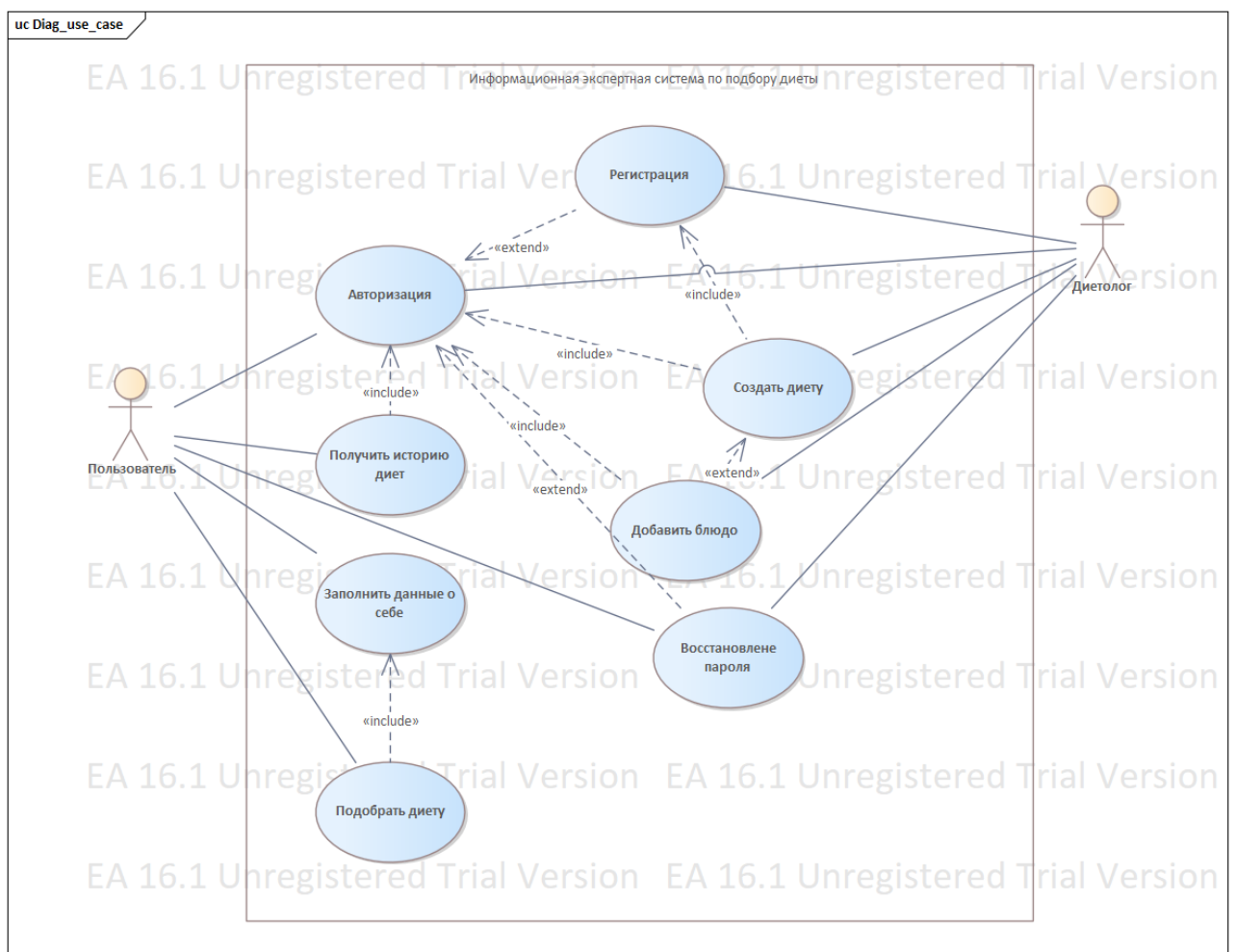


Рисунок 1

### 2. Описание модели анализа требований

#### а. Трассировка прецедентов в кооперации

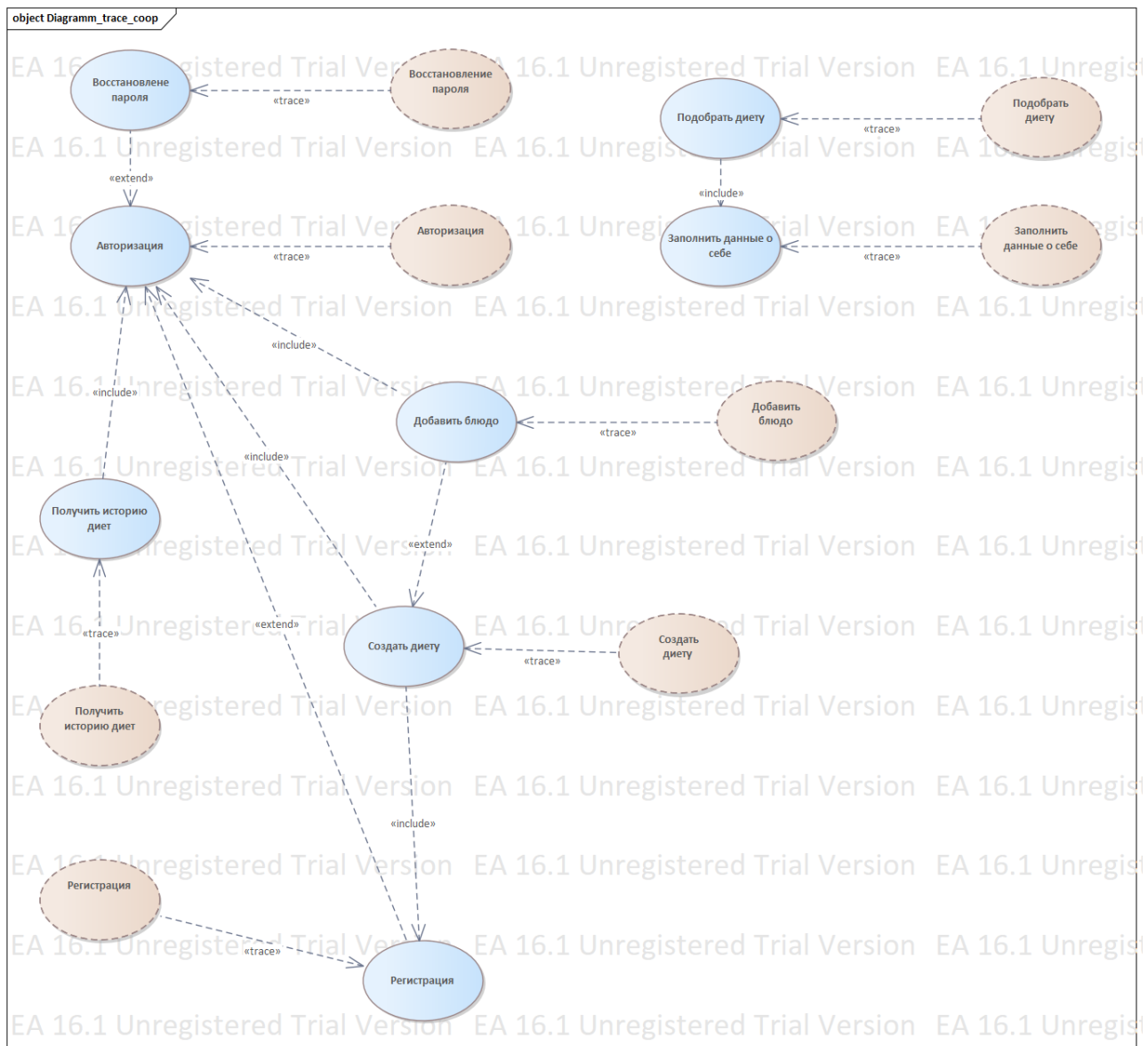


Рисунок 2

**в. Обзорные диаграммы классов сущностей, управляющих и граничных**

## Диаграмма классов сущностей:

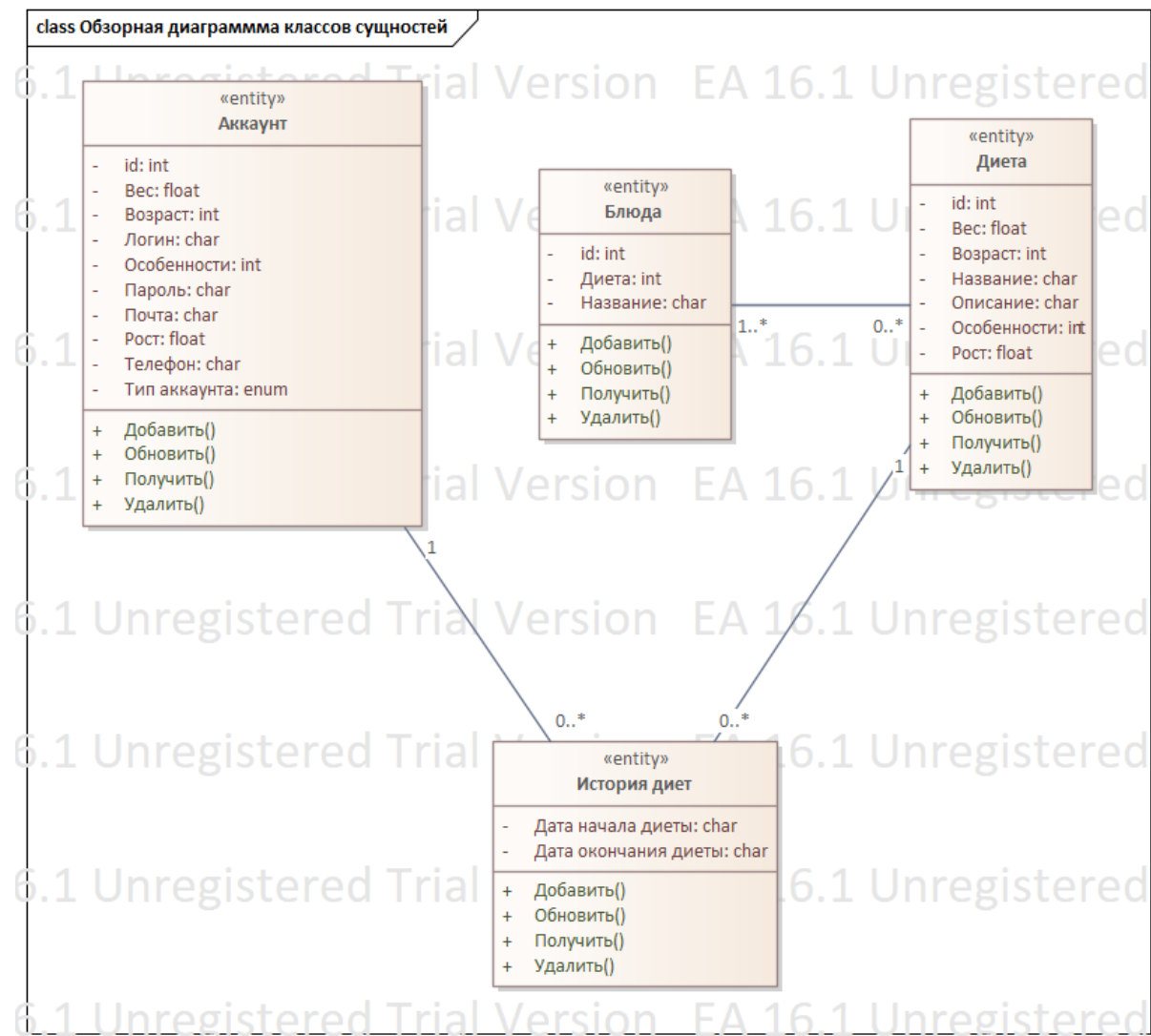


Рисунок 3

## Диаграмма классов управляющих:

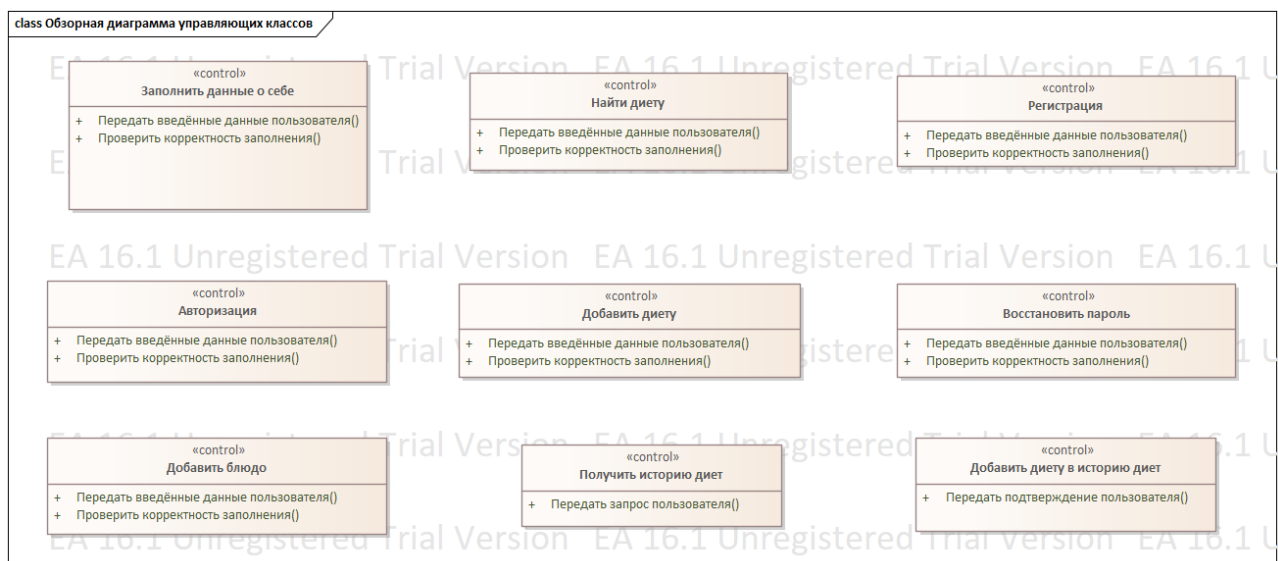


Рисунок 4

## Диаграмма классов граничных:



Рисунок 5

## с. Диаграмма последовательностей

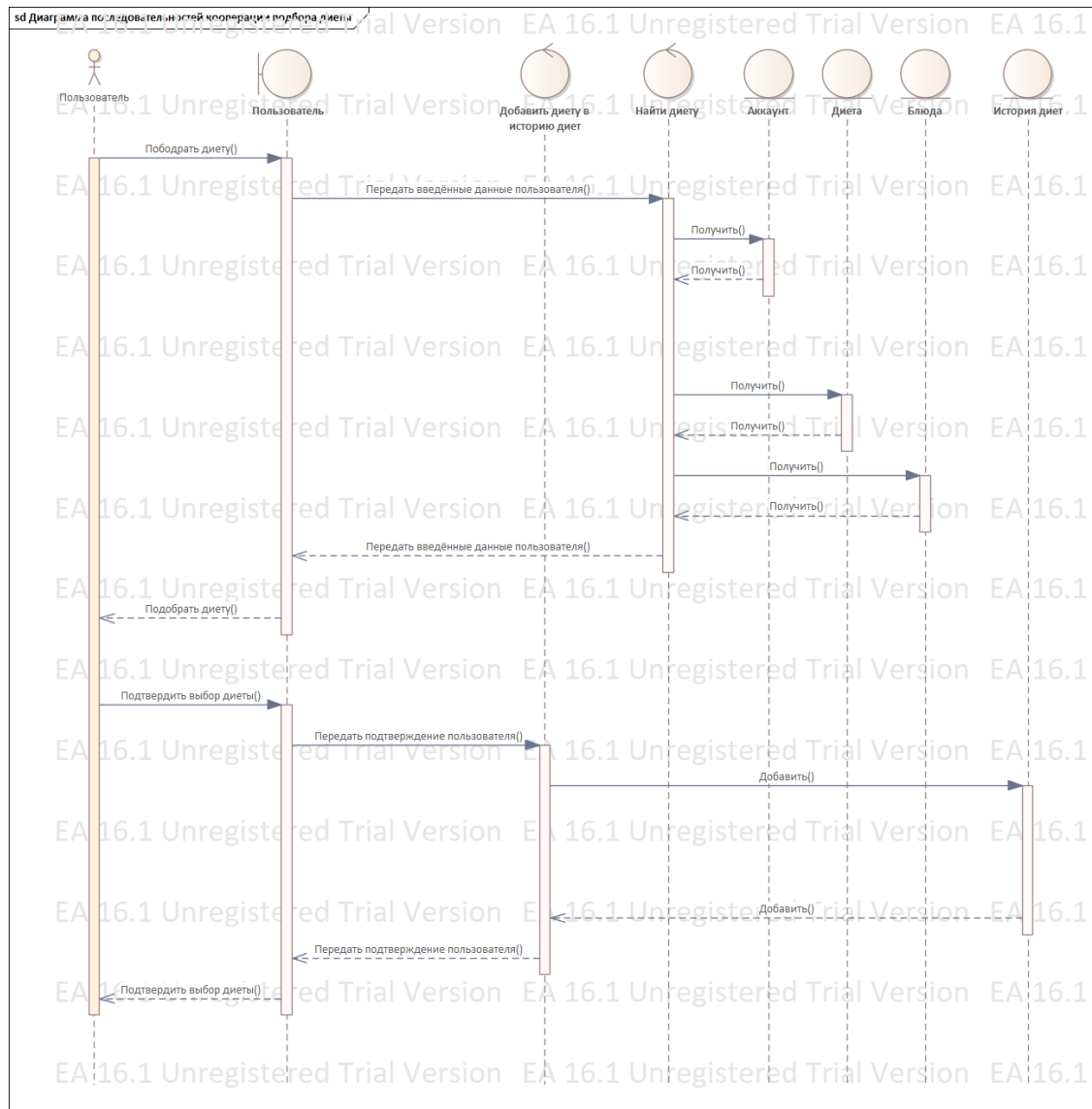


Рисунок 6

## d. Диаграмма классов кооперации подбора диеты

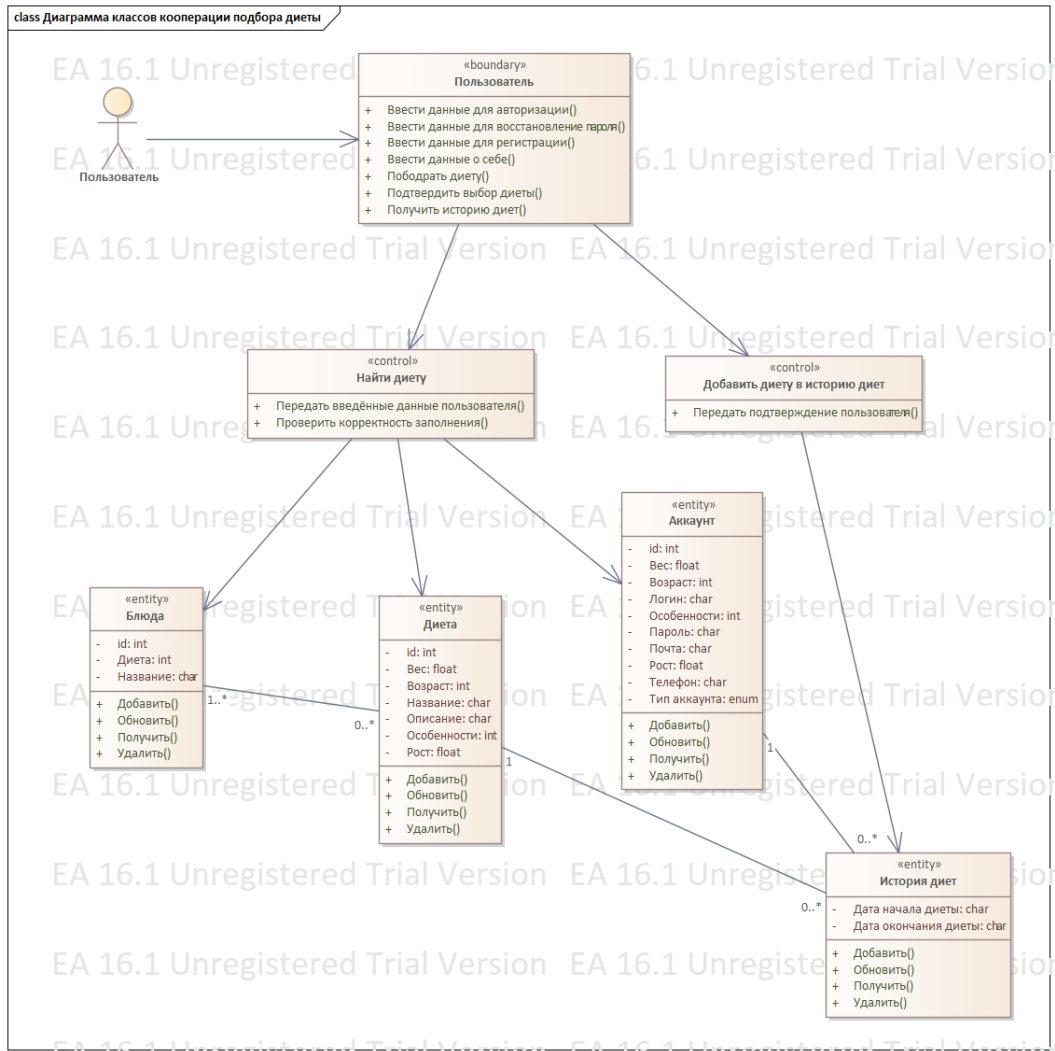


Рисунок 7

## е. Обзорная диаграмма всех классов анализа

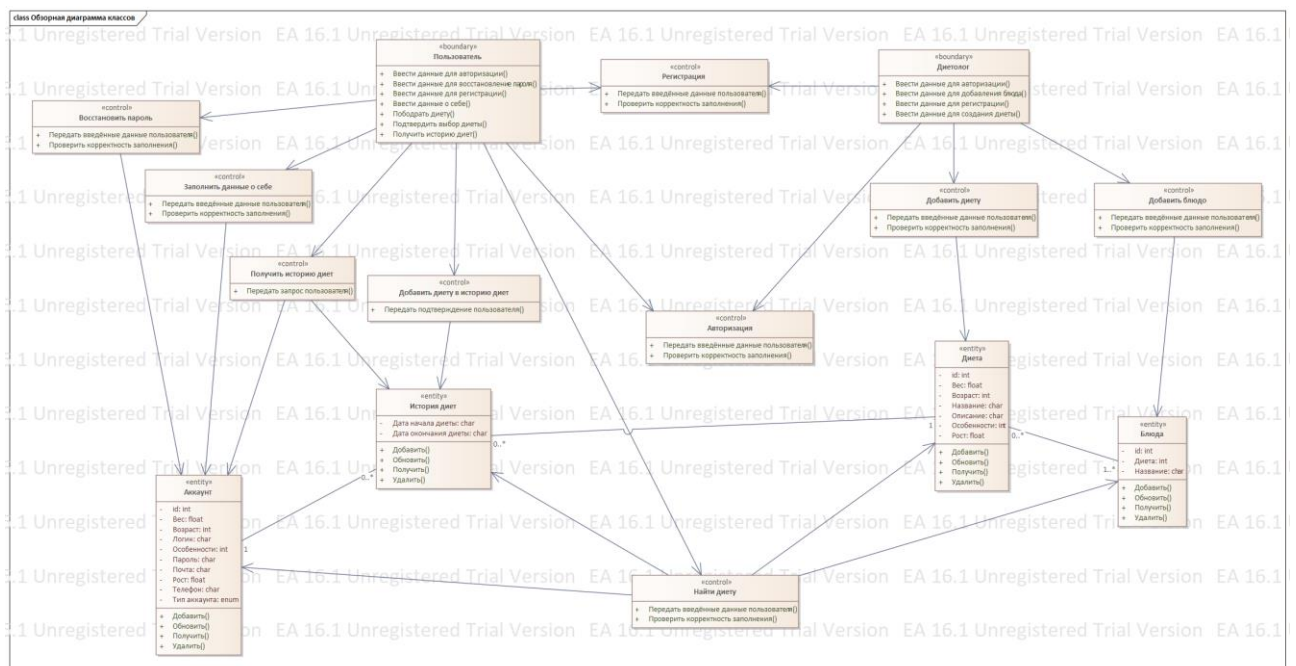


Рисунок 8

## ф. Диаграмма пакетов



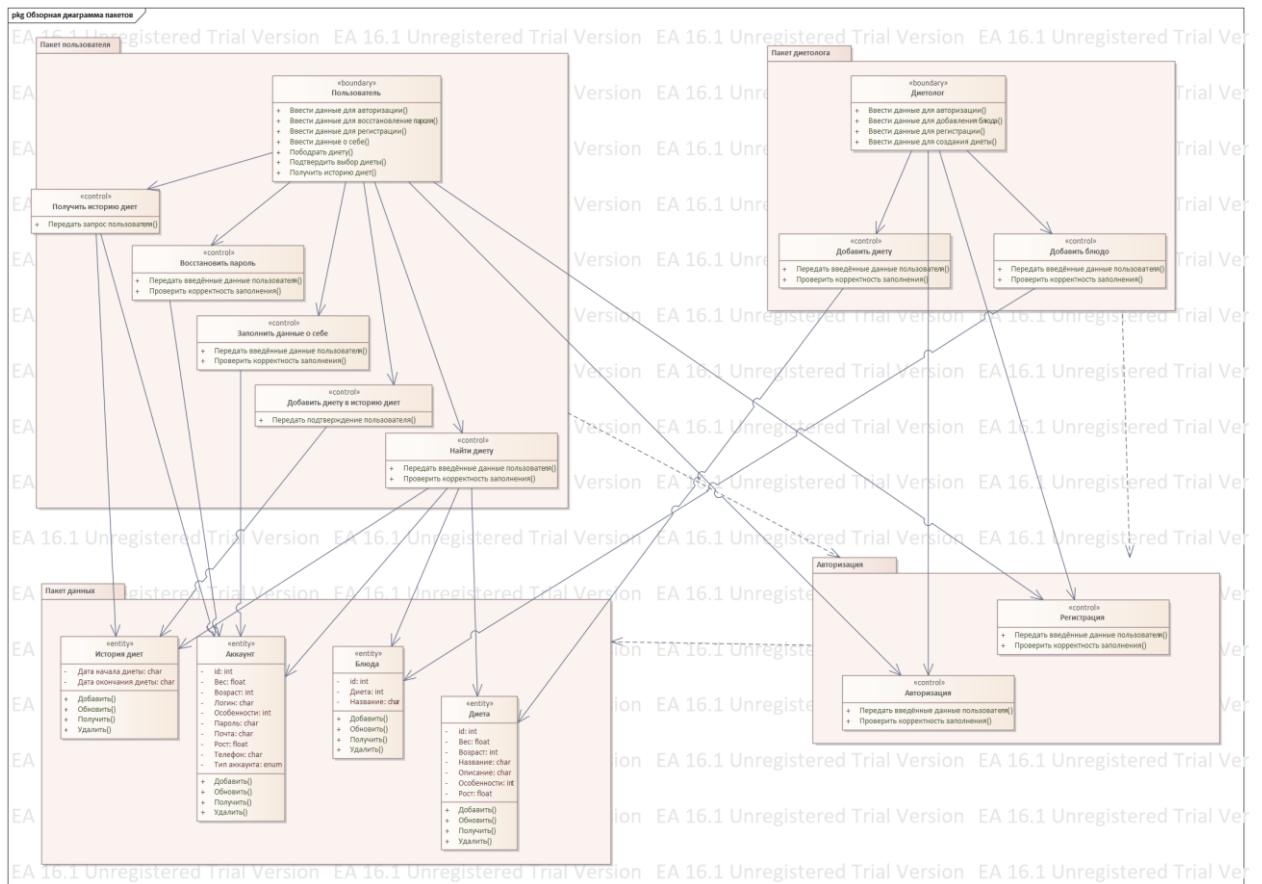


Рисунок 9

## Выводы:

В ходе выполнения лабораторной работы был изучен унифицированный процесс разработки (RUP); приобретены умения построения модели анализа и получены навыки построения модели требований в среде Enterprise Architect.

## Список источников

1. Sparx Systems – Текст. Изображение.: электронные // Sparx Systems : [сайт]. – URL: <https://sparxsystems.com/> (дата обращения 15.06.2022)
2. Виноградова, М. В. Унифицированный процесс разработки программного обеспечения: учебное пособие / М. В. Виноградова, В. И. Белоусова. — Москва: Издательство МГТУ им. Н.Э. Баумана, 2015. — 80, [2] с.: ил. ISBN 978-5-7038-4265-2
3. Якобсон А., Дуч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. / А. Якобсон, Г. Дуч, Дж. Рамбо. – Спб.: Питер. – 2002.
4. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. – Пер. с англ. – СПб: СимволПлюс, 2007. – 624 с., ил. ISBN13: 9785932860946 ISBN10: 5932860944
5. Руководство пользователя Enterprise Architect 15.1 – Текст. Изображение.:электронные //SparxSystems [https://sparxsystems.com/enterprise\\_architect\\_user\\_guide/15.1/index/index.html](https://sparxsystems.com/enterprise_architect_user_guide/15.1/index/index.html) (дата обращения 15.06.2022)



6. Технологии разработки программного обеспечения: Учебник/ С. Орлов.  
— СПб.:Питер, 2002. — 464 с.: ил. ISBN 5-94723-145-X