*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***к курсовому проекту по дисциплине   
"Технологии программирования"***

***на тему***

*Проектирование и разработка программной системы   
информационной системы «Университет»*

*Выполнил: Грачев Д. А.*

*Принял: доц. кафедры ИСПИ*

*Верешинин В.В.*

*Владимир, 2022*

**<утвержденный Лист задания нА курсовой проект>**

**Аннотация**

Аннотация составляется на русском и английском языках. Это краткая характеристика работы, ее содержимого. В конце указывается свойства документа (пояснительной записки).

Курсовой проект представлен на 56 страницах, рисунков – 5, таблиц – 11, использованных источников – 4, приложений – 2, иллюстрационный материал на 3 листах формата А1.

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc116028136)

[1 Описание предметной области 4](#_Toc116028137)

[1.1. Описание предметной области 4](#_Toc116028138)

[1.2. Пользователи разрабатываемой системы 4](#_Toc116028139)

[1.3. Словарь предметной области 5](#_Toc116028140)

[2 проектирование системы 6](#_Toc116028141)

[заключение 10](#_Toc116028142)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 11](#_Toc116028143)

[ПРИЛОЖЕНИЕ А Исходный код программной системы 12](#_Toc116028144)

[ПРИЛОЖЕНИЕ Б Структура таблиц БД 13](#_Toc116028145)

# ВВЕДЕНИЕ

# Описание предметной области

* 1. Описание предметной области

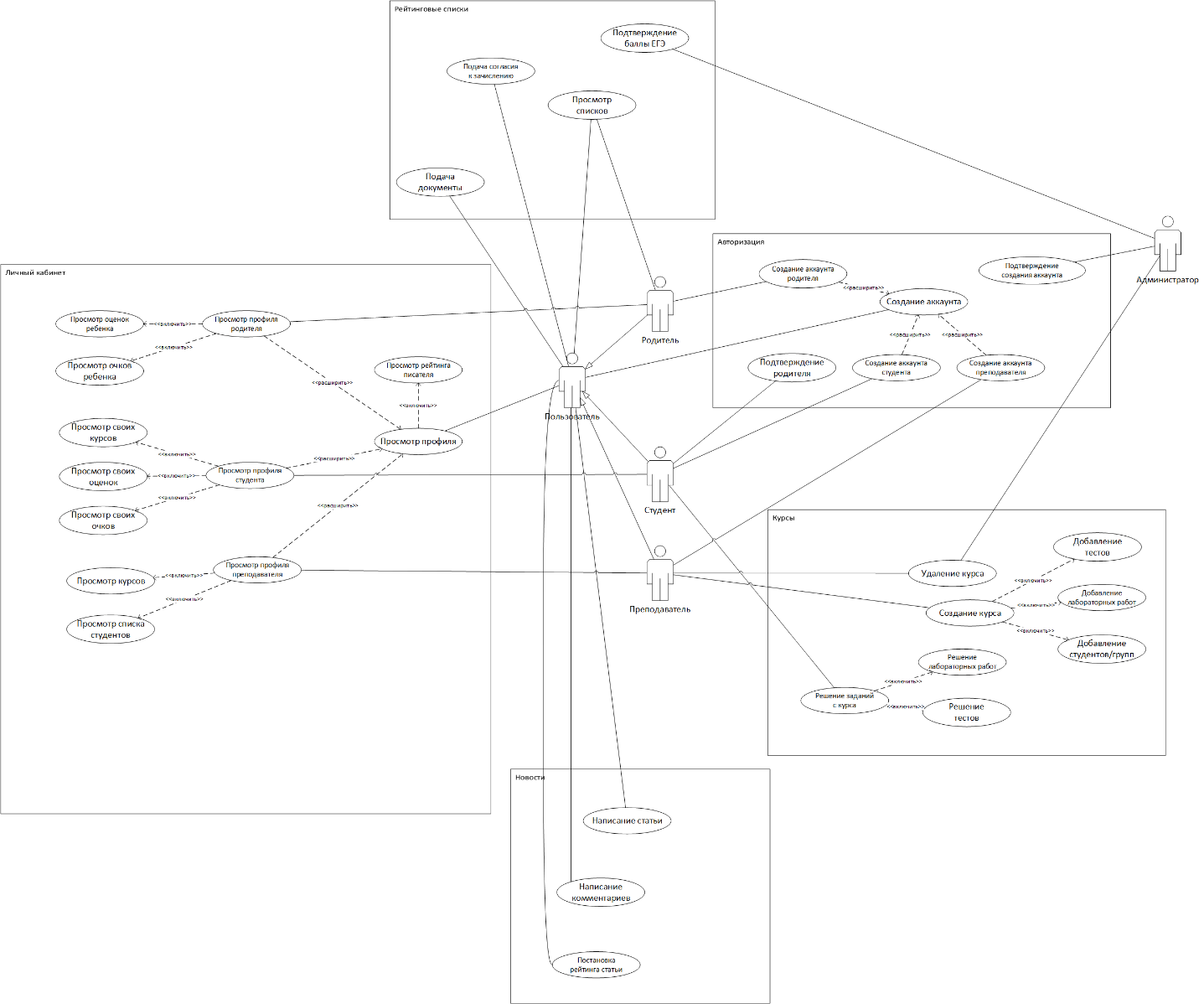
В университете, обучающем студентов и осуществляющем набор абитуриентов, важно иметь программную систему, обеспечивающую:

1. автоматизацию рейтинговых списков абитуриентов
2. сдачу лабораторных работ в электронной форме и контрольных работ в онлайн формате с автоматическим указанием баллов
3. новостную ленту с возможностью написания статей любым пользователем программной системы
   1. Пользователи разрабатываемой системы
4. Администратор – сотрудник университета, который администрирует работу программной системы, подтверждает некоторые действия пользователей
5. Пользователь – зарегистрированный участник системы, который может полностью взаимодействовать с подсистемой новостей и подавать документ на поступление в университет
6. Студент – зарегистрированный участник системы, который владеет всеми возможностями обычного пользователя, так же может взаимодействовать с подсистемой курсов
7. Родитель – зарегистрированный участник системы, который владеет всеми возможностями обычного пользователя, так же может привязываться к аккаунту студента, после подтверждения этого последним, для просмотра успеваемости ребенка
8. Преподаватель – зарегистрированный участник системы, который обладает всеми возможностями обычного пользователя, так же может взаимодействовать с подсистемой курсов
   1. Словарь предметной области
9. *Курс* – часть программной системы, с помощью которой «Преподаватель» может создавать задания для «Студента», такие как лабораторные и контрольные работы
10. *Новостная лента* – часть программной системы, обеспечивающая возможность написания новостей любым пользователем, так же любой прочитавший новость может поставить оценку прочитанному материалу и оставить комментарий
11. *Рейтинговые списки* – часть программной системы, обеспечивающая автоматизацию работы с абитуриентами
12. *Личный кабинет* – часть программной системы, внутри которой пользователь может увидеть данные о своем аккаунте, такие как рейтинг писателя, курсы и оценки, если это аккаунт студента, курсы, если это аккаунт преподавателя, оценки ребенка, если это аккаунт родителя

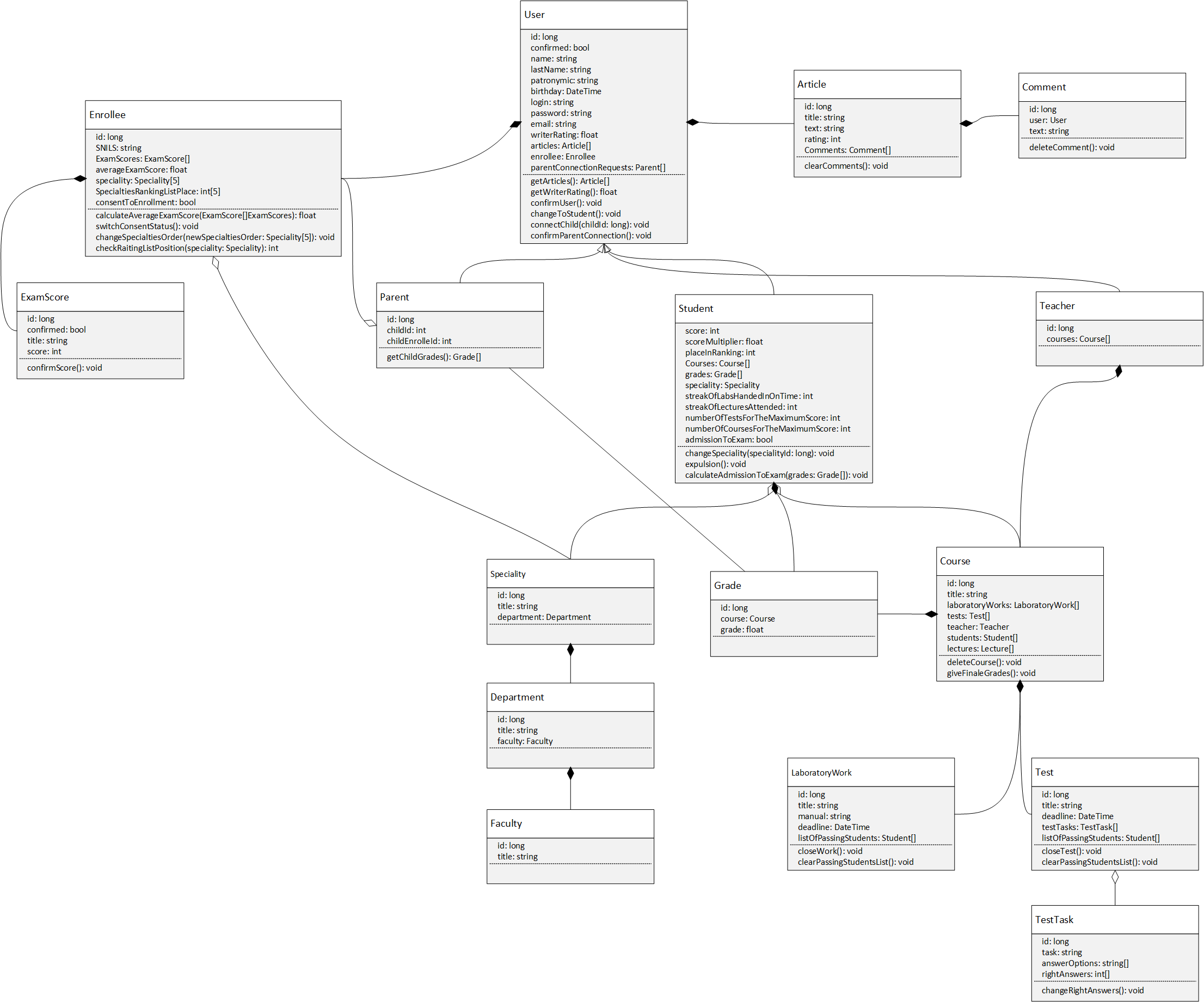
# проектирование системы

Разработанные диаграммы:

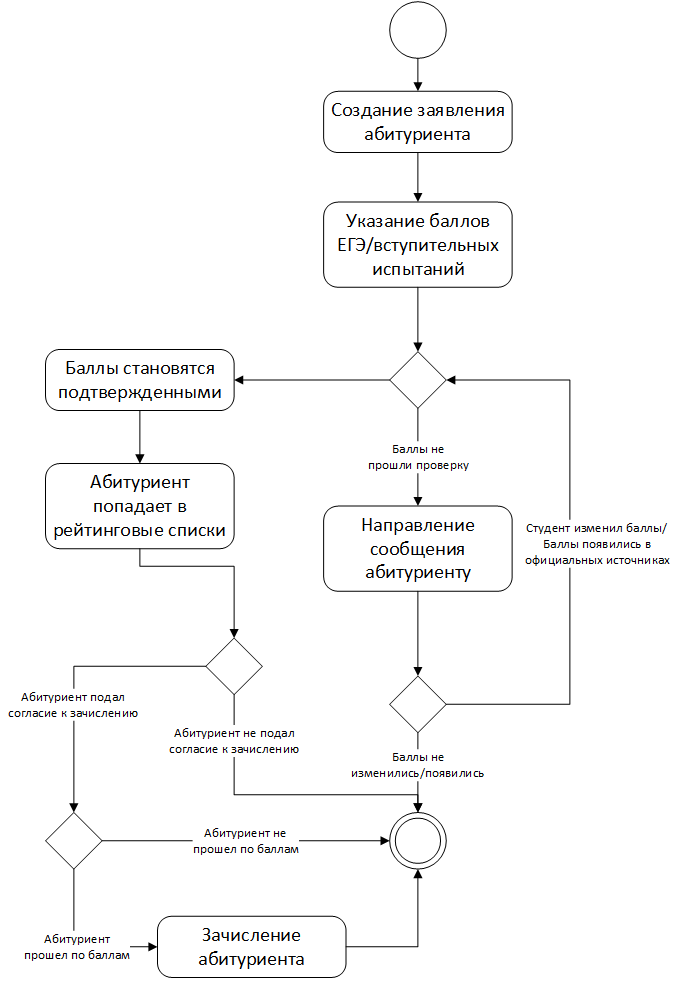
1. Диаграмма прецендентов



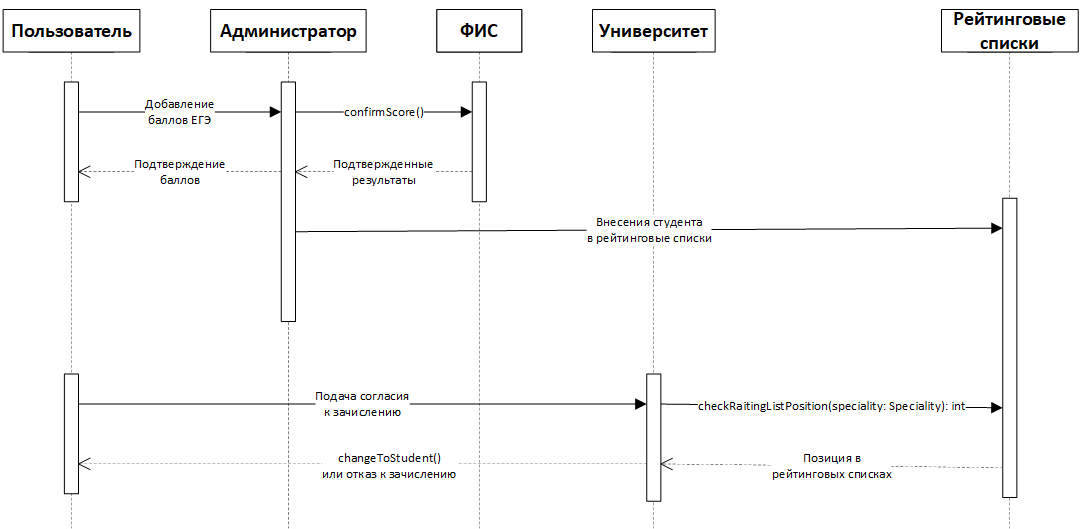
1. Диаграмма классов



1. Диаграмма состояния



1. Диаграмма последовательностей



1. Спецификация прецендента «Создание аккаунта»
   1. **Название:** «Создание аккаунта»
   2. **Предусловие:** Пользователь зашел на сайт и нажал кнопку регистрации
   3. **Действующее лицо:** Пользователь
   4. **Основной поток:** Пользователь открывает форму, вводит свои данные  
      Данные пользователя сохраняются в базе данных, и пользователь регистрируется
   5. **Альтернативный поток:** такой пользователь уже существует  
      Пользователю поступает информация о том, что такой пользователь уже существует и будет предложено изменить данные
   6. **Постусловие:** Если пользователь создался, то он будет авторизирован в системе
2. Спецификация прецендента «Написание статьи»
   1. **Название:** «Написание статьи»
   2. **Предусловие:** Авторизированный пользователь зашел на страницу создания новости
   3. **Действующее лицо:** Пользователь
   4. **Основной поток:** Пользователь пишет статью, добавляет картинки
   5. **Альтернативный поток:** Если при отправке статьи возникнет ошибка, она будет выведена диалоговым окном пользователю
   6. **Постусловие:** Если пользователь дописал статью, то он ее может опубликовать для остальных пользователей
3. Спецификация прецендента «Решение заданий с курса»
   1. **Название:** «Решение заданий с курса»
   2. **Предусловие:** Преподаватель создал курс, в который добавил задания и присоединил студентов
   3. **Действующее лицо:** Студент
   4. **Основной поток:** Студент заходит в курс и открывает задание, выполняет его  
      Получает оценку из 5 баллов
   5. **Альтернативный поток:** Получил оценку меньше 2  
      В зависимости от настроек задания студент может пере пройти тест
   6. **Постусловие:** Если студент прошел полностью тест и получил оценку, оценка добавляется в его ЛК

# заключение

Основные выводы по работе и достигнутые результаты. Достаточно трех четырех абзацев текста.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Липаев В.В. Проектирование программных систем. М.: Высш. шк, 1990.

2. Буч Г. Объектно-ориентированное проектирование / Пер. с анг. Конкорд, 1996.

3. Майерс Г. Надежность программного обеспечения. М.: Мир, 1980

4. <название статьи/книги/материала> [Электронный ресурс]: <автор>. – Режим доступа: <полный URL>

ПРИЛОЖЕНИЕ А Исходный код программной системы

package ru.home.lt.cf;

import lrapi.lr;

import org.apache.commons.configuration.ConfigurationException;

import org.apache.commons.configuration.XMLConfiguration;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class UC01CreateApplication extends UseCase{

final static private Logger logger = LoggerFactory.getLogger(UC01CreateApplication.class);

public static Double createILSFPercentage = 82.82;

public static Double createCWFPPercentage = 10.80;

public static Double createLLQ1Percentage = 5.21;

public static Double createCORNPercentage = 0.78;

public static Double createCCLPPercentage = 0.39;

private UC01CreateILSF ilsf;

private UC01CreateCWFP cwfp;

private UC01CreateLLQ1 llq1;

private UC01CreateCORN corn;

private UC01CreateCCLP cclp;

static {

XMLConfiguration config = null;

try {

config = new XMLConfiguration("../load-test.xml");

} catch (ConfigurationException e) {

logger.error(e.getLocalizedMessage());

}

createILSFPercentage = config.getDouble(LoadTestConfiguration.CREATE\_ILSF\_APPLICATION\_PERCENTAGE, 82.82);

createCWFPPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CWFP\_APPLICATION\_PERCENTAGE, 10.80);

createLLQ1Percentage = config.getDouble(LoadTestConfiguration.CREATE\_LLQ1\_APPLICATION\_PERCENTAGE, 5.21);

createCORNPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CORN\_APPLICATION\_PERCENTAGE, 0.78);

createCCLPPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CCLP\_APPLICATION\_PERCENTAGE, 0.39);

LogProductPercentages();

}

ПРИЛОЖЕНИЕ Б Структура таблиц БД



Рисунок Б.1. Структура таблицы "Пользователи"

**карман с диском на котором записаны материалы курсового проектирования:**

1. пояснительная записка

2. проект системы с исходными кодами

3. дополнительные компоненты системы, необходимые для ее развертывания

4. прочие вспомогательные источники

5. презентация по которой будет проходить защита работы

**Папка(файлик) для графического материала к пояснительной записке и сам графический материал в ней**