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**

**«Распределенные программные системы»**

**на тему**

Проектирование и разработка программной системы

информационной системы «Дистанционное обучение студентов».

Выполнили: студент гр. ПРИ-120

Ермолаев Д. А. и Казаков К. А.

Принял: Тимофеев А. А.

Владимир, 2022

**Этап 1**

**Постановка задачи.**

Целью разработки данного курсового проекта является разработка программной системы «Дистанционное обучение студентов», включающая в себя подсистему хранения данных и позволяющая студентам выполнять различные задания, которые оцениваются лекторами.

Функциональные требования:

* Авторизация и аутентификация;
* Создание справочников студентов;
* Редактирование справочников студентов;
* Создание справочников лекторов;
* Редактирование справочников лекторов;
* Создание различных ролей;
* Создание тестов;
* Редактирование тестов;
* Прохождение тестов;
* Предоставление студентам доступа к курсу;
* Включение студентов в дистанционное прохождение тестов;
* Оценка тестов;
* Вывод результатов тестов;

Нефункциональные требования;

* База данных MS SQL;
* Использование SpringFramework;

**Описание предметной области.**

В данной работе рассматривается предметная область «Дистанционное обучение студентов» предназначенная для обучения студентов вне учебного заведения.

Все студенты распределены по группам относительно направлению подготовки. Также они имеют свои личные данные.

Лекторы могутсоздавать дисциплины, добавлять в них учебные материалы, лабораторные работы, создавать тесты. К этим дисциплинам они могут подключить определенные группы студентов. В свою очередь студенты могут выполнять созданные лектором задания.

Администратор разрабатываемой системы может создавать и редактировать аккаунты для студентов и лекторов, чтобы те впоследствии могли авторизоваться на электронном сайте и пользоваться им.

**Пользователи системы.**

**Студент** – обучающийся, который может выполнять задания и просматривать лекции.

**Лектор**– преподаватель, имеющий возможность создавать и редактировать дисциплины.

**Администратор**– лицо, администрирующее электронный сайт, а также имеющее возможность создавать и редактировать аккаунты студентов и лекторов.

**Словарь предметной области.**

*Тест* – задание, выполняемое студентом, для проверки теоретических знаний.

Атрибуты: название, дата сдачи, список заданий, баллы.

*Задание –* вопрос, содержащийся в тесте, включающий в себя варианты ответа.

Атрибуты: название, список вариантов ответа.

Вариант ответа – одно из нескольких решений задания.

Атрибуты: название.

*Лабораторная работа* – задание, выполняемое студеном, содержащее в себе методические указания.

Атрибуты:название, дата сдачи, методические указания, баллы.

*Группа* – состав студентов, относящихся к определенному направлению подготовки.

Атрибуты: название, направление подготовки, список дисциплин.

*Дисциплина* – предмет, включающий в себя, лабораторные работы, тесты и методические материалы.

Атрибуты: название,список тестов, лектор, список студентов, лабораторных работ.

*Пользователь* – человек, использующий функционал автоматизированной электронная системы.

Атрибуты: ФИО, логин, пароль, e-mail, дата рождения, номер телефона.

**Описание основных прецедентов пользователей в системе:**

1. **Название: «Создать дисциплину»**

*Предусловие:* Пользователь авторизован

*Действующее лицо:*Лектор

*Основной поток:*Создание дисциплины

Лектор открывает электронный сайт (представление веб-приложения), переходит на страницу создания дисциплины. Далее может добавить лабораторные работы, тесты и методические указания. Также может предоставить доступ к дисциплине студентам. Нажимает на кнопку создания дисциплины.

Система сохраняет в базу данных созданную дисциплину.

*Альтернативный поток:*Дисциплина, которую лектор хочет создать, уже существует.

Выводится уведомление об этом.

*Постусловие:*После создания дисциплины появляется окно оповещения о создании.

1. **Название: «Вывести результаты теста»**

*Предусловие:* Пользователь авторизован

*Действующее лицо:* Студент

*Основной поток:*Вывод результатов теста

Студент открывает электронный сайт (представление веб-приложения), переходит на страницупройденного теста, на которой выводятся оценка, поставленная лектором (прецедент: «Оценка теста»).

*Альтернативный поток:* Тест не оценён.

В окне, в котором прописывается оценка, указано: «тест не оценён».

1. **Название: «Оценка теста»**

*Предусловие:* Пользователь авторизован, студент должен пройти тест

*Действующее лицо:* Лектор

*Основной поток:*Оценка теста

Лектор открывает электронный сайт (представление веб-приложения), переходит на страницу, на которой отображаются тесты, пройденные студентами, выбранной дисциплины. Проверяет и выставляет оценку.

Система сохраняет в базу данных количество баллов за тест.

*Постусловие:*После выставления оценки появляется окно оповещения об оценивании.

1. **Название: «Создать лектора»**

*Предусловие:* Пользователь авторизован

*Действующее лицо:* Администратор

*Основной поток:*Создание лектора

Администратор открывает электронный сайт (представление веб-приложения), переходит на страницу создания пользователей. Далее он может создать новый аккаунт и выдать ему права лектора.

Система сохраняет в базу данных информацию о созданном лекторе.

*Постусловие:*После создания лектора появляется окно оповещения о создании.

1. **Название: «Авторизоваться на сайте»**

*Предусловие:* Пользователь не авторизован

*Действующее лицо:* Пользователь

*Основной поток:*Авторизация на сайте

Пользователь открывает электронный сайт (представление веб-приложения), переходит на страницуавторизации. Далее вводит логин и пароль.

*Альтернативный поток:*Логин или пароль указаны неверно.

*Постусловие:*После авторизации пользователь получает доступ к возможностям сайта.

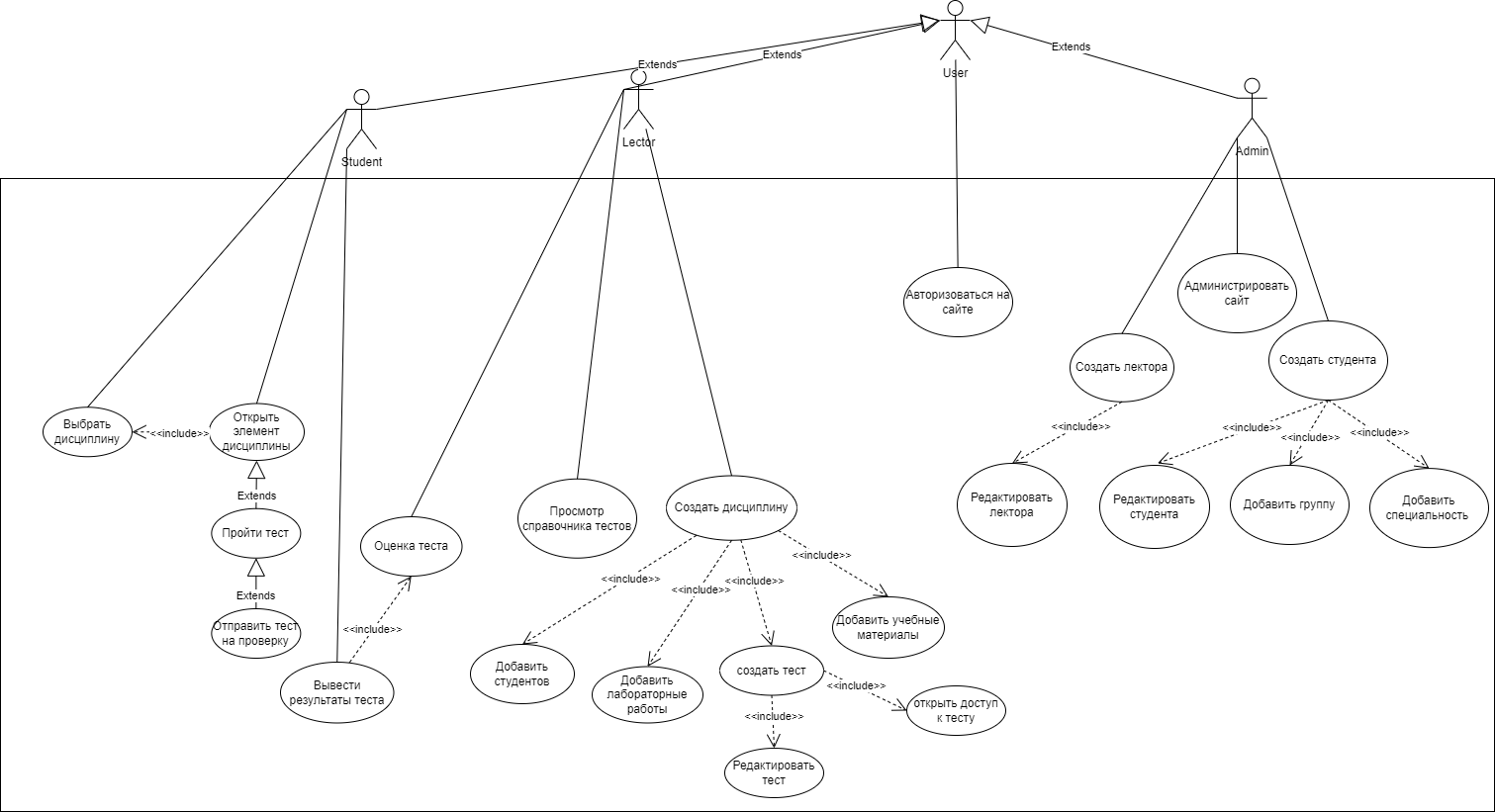


Рис. 1. Диаграмма прецедентов.

Разработка **диаграммы классов (UMLClassDiagram)** в нотации UML.

Класс – это описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой.

Выделим такие основные сущности в системе, как:

- **Student** (пользователь, обладающий правами студента)

- **Group** (список студентов, которые учатся по своей специальности)

- **Lector** (пользователь, обладающий правами лектора)

-**LabWork** (задания, имеющие методические указания, выполняемые студентом)

-**Discipline** (предмет, который содержит в себе лабораторные работы, тесты и учебные материалы)

- **Test** (задания по итогам тем(ы), выполняемые студентом)

- **User** (абстрактный пользователь с его данными в системе)

- **Task** (задание, содержащееся в тесте)

- **AnswerOption** (один из вариантов ответа, содержащийся в задании)

Используются такие основные виды отношений, как:

* Наследование:

1. User: Student, Lector.

* Композиция:

1. LabWork - Student \* - \*;
2. Group - Student1 - \*;
3. Discipline - Group\* - \*;
4. Discipline – Lector \* - 1;
5. LabWork – Discipline \* - 1;
6. Test – Discipline \* - 1;
7. Test – Student \* - \*;
8. Task – Test \* - 1;
9. AnswerOption – Task \* - 1

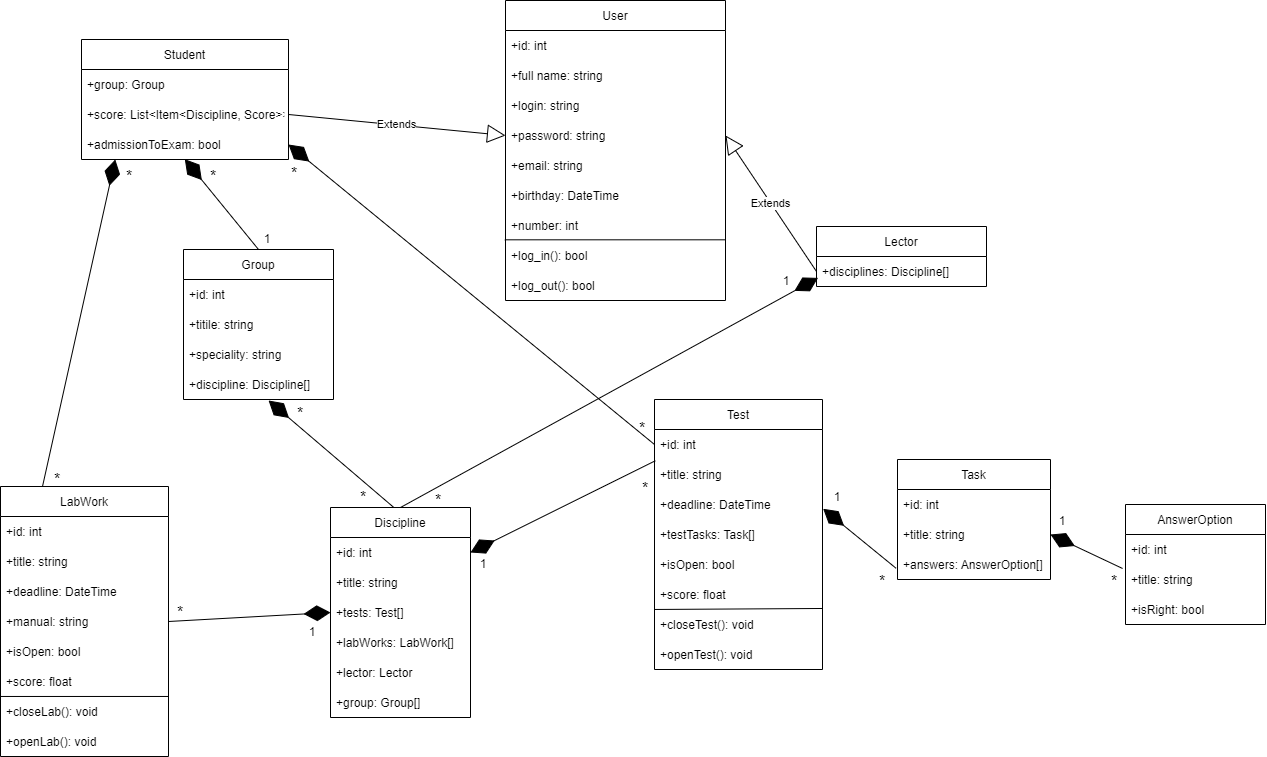
****

Рис. 2. Диаграмма классов.

Далее создадим набросок будущего интерфейса системы (Рис. 3).

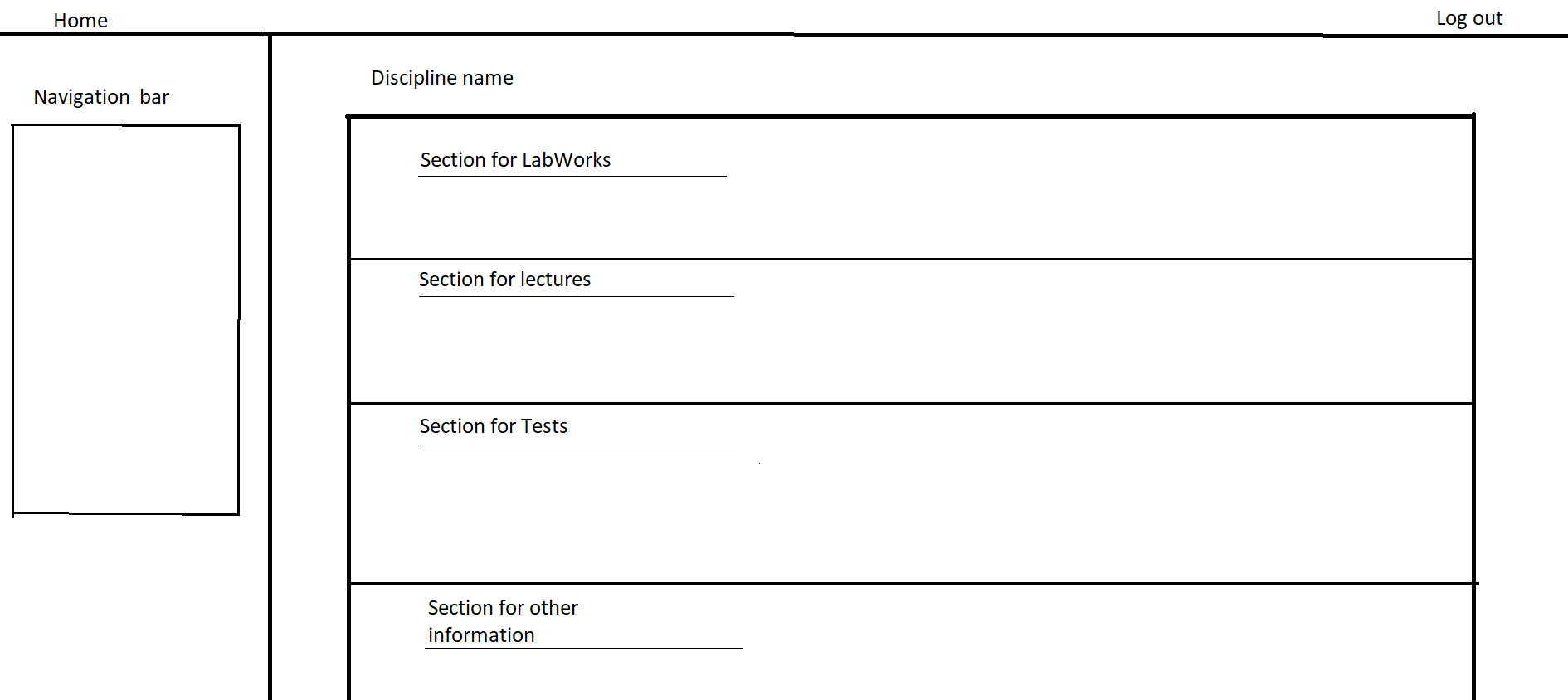


Рис. 3. Набросок интерфейса.