**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ИМ. И.РАЗЗАКОВА**

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ СИСТЕМ**

Техническое задание

на разработку программы:

«Тестирование»

**Выполнили:** студенты 3-го курса, группы: ПИ-1-18

Цой Кристина, Калашникова Полина

И студент 2-го курса

 Улан у. Нурдин

Бишкек 2021

Оглавление

[1. **Введение** 3](#_Toc75885826)

[**2.** **Актуальность темы** 3](#_Toc75885827)

[**3.** **Назначение и цели создания приложения** 4](#_Toc75885828)

[3.1 Назначение программы: 4](#_Toc75885829)

[3.2 Цель разработки: 4](#_Toc75885830)

[**4.** **Технические требования** 5](#_Toc75885831)

[**4.1** **Функциональные требования:** 5](#_Toc75885832)

[4.1.1 Диаграмма вариантов использования (Use Case) 5](#_Toc75885833)

[**4.2** **Нефункциональные требования** 6](#_Toc75885834)

[4.2.1 Требования к информационным структурам и методам решения 6](#_Toc75885835)

[4.2.2 Требования к исходным кодам и языкам программирования 6](#_Toc75885836)

[4.2.3 Требование к надежности 6](#_Toc75885837)

[4.2.4 Требования к информационной и программной совместимости 7](#_Toc75885838)

[4.2.5 Требования к программной документации: 7](#_Toc75885839)

[**5.** **Обзор аналогов** 8](#_Toc75885840)

[**6.** **Проектирование** 10](#_Toc75885841)

[6.1 Контекст моделирования 10](#_Toc75885842)

[6.1.1 Диаграмма бизнес-процессов (IDEF0) 10](#_Toc75885843)

[6.1.2 Диаграмма потоков данных (DFD) 12](#_Toc75885844)

[6.2 Диаграмма ERD (ER-Модель) 14](#_Toc75885845)

[6.3 Диаграмма классов (Сlass diagram) 15](#_Toc75885846)

[6.4 Диаграммы деятельности (Activity diagram) 17](#_Toc75885847)

[6.5 Диаграмма последовательности (Sequence diagram) 20](#_Toc75885848)

[**7.** **Заключение** 24](#_Toc75885849)

[**8.** **Литература** 24](#_Toc75885850)

## **Введение**

В ходе образовательного процесса особое внимание следует уделять контролю усвоенного учащимися материала. Из всех способов контроля за уровнем обученности студентов наиболее перспективным, является тестирование. Т. к, оно обеспечивает преподавателю объективную и оперативную информацию об уровне усвоения студентами обязательного учебного материала, а администрацию вуза информирует об объективности управления. Также, результаты тестирования – это показатель качества преподавания. Система компьютерного тестирования –– это универсальный инструмент для определения обученности студентов на всех уровнях образовательного процесса. Единая система тестирования позволяет комбинировать тестовые задания разных дисциплин в рамках одного блока или дисциплин одной тематической направленности. Результаты текущего и рубежного тестирования –– это не только объективный показатель освоения студентами темы, раздела или дисциплины, но и, прежде всего, показатель качества работы преподавателя.

## **Актуальность темы**

В университетах проводится тестирование нескольких типов:

* Междисциплинарная итоговая государственная аттестация (история, география, кыргызский язык 1/2);
* Государственный экзамен по направлению подготовки (спец.предметы).
* Модульное тестирование (проводиться каждый семестр по всем предметам)

Вручную проверять такое количество работ и рассчитывать баллы довольно трудоёмкая и время затратная задача, а также высока вероятность допустить ошибку либо в проверке, либо в расчетах. Тяжело отлёживать задолжников, по всем факультетам. А на регистрацию каждого студента уходит много времени. Что приводит к накладкам во время тестирования. Из-за вышеперечисленного появилась необходимость в программе, которая обеспечит тестирование.

*Примечание: до государственного тестирования не допускаются студенты, у которых присутствуют задолженности по* *предметам (история, география и кыргызский язык* *1/2). Программа должна это учитывать. Также до модульного тестирования не допускаются если есть задолженность по оплате или не допуск по дисциплины.*

## **Назначение и цели создания приложения**

* 1. Назначение программы:
* Отслеживание задолжников по дисциплинам
* Регистрация студентов на тестирование, если у таковых нет задолженностей по дисциплинам (событие при Государственном междисциплинарном экзамене)
* Регистрация всех студентов даже если есть задолженности по спец предметов (событие при Государственном экзамене по направлению подготовки)
* Регистрация студентов если у них задолженность по оплате и допуск по предмету (событие при Государственном экзамене по направлению подготовки)
* Тестирование студентов
* Подсчет баллов после завершения тестирования.
* Выставление итоговых баллов в электронную ведомость студента
* Отслеживание времени вовремя тестирования
* Перерегистрация студента на 7-8 семестр, не сдавших Гос. Экзамен
* Объективное измерение и интерпретация результатов
  1. Цель разработки:
* Увеличить скорость регистрации студентов на экзамен
* Отслеживать всех задолжников по факультетам
* Исключить ошибки из-за человеческого фактора
* Проверить знания студентов
* Показать качество преподавания в университете
* Автоматизировать обработку результатов
* Предоставить баллы студентам и преподавателям

## **Технические требования**

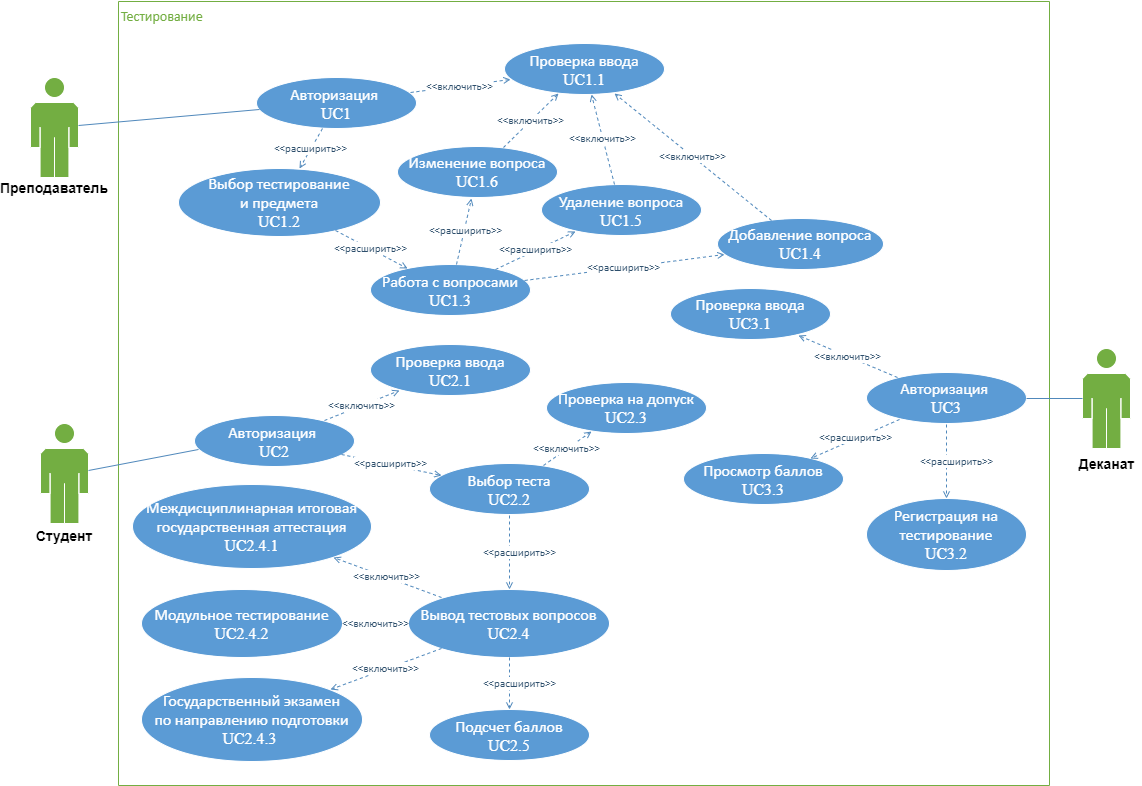
### **Функциональные требования:**

Система должна включать в себя следующие функции :

* Регистрировать и хранить следующие справочные данные:
* Данные об успеваемости студентов;
* Тестовые вопросы;
* Количество баллов за каждый вопрос
* Отслеживать задолжников, т.е. студентов, у которых есть задолженности по предметам (история, география и кыргызский язык1/2);
* Отслеживать задолжников по оплате
* Сохранять баллы за тестирование и обеспечивать возможность просмотра баллов;
* Фильтровать данные по следующим критериям:
* Факультет;
* Направление;
* Группа;
* Определенный студент.
  + 1. Диаграмма вариантов использования (Use Case)

Следующая диаграмма Use Case представлена на **рис. 1** демонстрирует сценарий взаимодействий деканата, преподавателя и студента с функциями программы.

**Диаграмма вариантов использования** (англ. use case diagram) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.



**Рис. 1 Use Case**

**Описание Use Case:**

### **Нефункциональные требования**

* + 1. Требования к информационным структурам и методам решения

База данных работает под управлением Microsoft SQL Server. Используется многопоточный доступ к базе данных. Необходимо обеспечить одновременную работу с программой с той же базой, данной модулей экспорта внешних данных.

* + 1. Требования к исходным кодам и языкам программирования

Разработка осуществляется в среде MS Visual Studio 2019. Язык программирования **Java Spring**.

* + 1. Требование к надежности
* защищённость серверов от внезапного отключения питания и перепадов напряжения
* устойчивость оборудования и программного обеспечения
* защита данных университета
* защита от потери критичных данных
* устойчивости к внутренним ошибкам
* устойчивость к некорректным действиям пользователей системы
* устойчивость к атакам из сети Интернет
  + 1. Требования к информационной и программной совместимости

**Серверная часть:**

* Процессор: как минимум частотой 1 ГГц или SoC (System on a Chip, т.е. однокристальные системы) или выше;
* Оперативная память: 1 ГБ (для 32-разрядных систем) или 2 ГБ (для 64-разрядных систем);
* Операционная система семейства Windows

**Клиентская часть для ПК:**

* ОС: MS Windows 7/8/10, Mac OS X.
  + 1. Требования к программной документации:
* Разрабатываемые функции программы должны быть задокументированы, и закомментированы.
* Разрабатываемая программа должна иметь справочное пособие.
* Руководство пользователя: полная инструкция по использованию программы.

В контексте данной программы нефункциональными требованиями, будут являться ограничения информационной системы управления сервисного центра в следующих этапах:

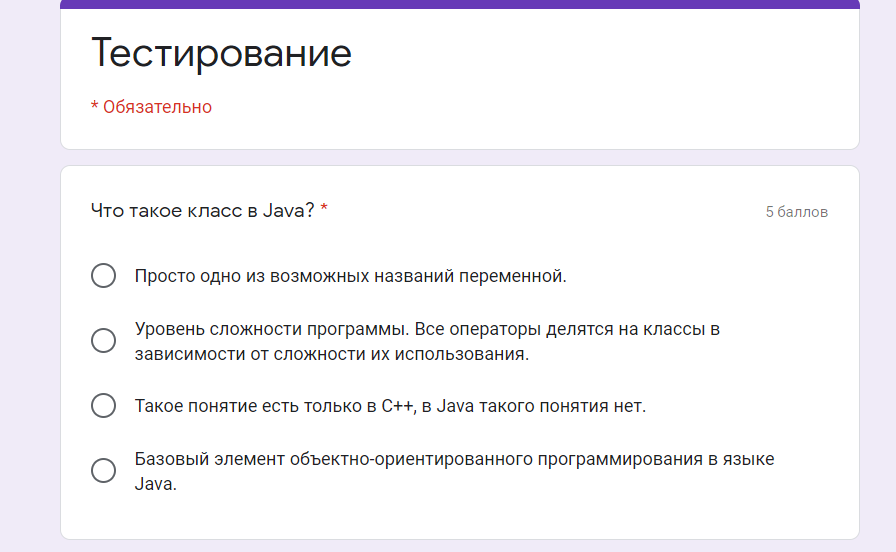
* Обязательное заполнение всех видов данных – это требование является обязательным к таким событиям, как например: логин и пароль, выделения хотя бы одного поля ответа в тестировании.
* Точность Программы – программа должна требовать максимально точные, и не противоречащие данные, как например: подсчет баллов после прохождения тестирование или аннулирование баллов если студент не успел в определенное время
* Удобство интерфейса – интерфейс должен быть максимально понятным для студента и сотрудника университета, которым он пользуется.

## **Обзор аналогов**

Рассмотрим несколько вариантов аналогов:

**Google Form**

Google Формы – отличный инструмент для создания онлайн тестов, которые могут использоваться для самых разных целей. C их помощью можно провести тестирование учеников или студентов и быстро оценить их уровень знаний.

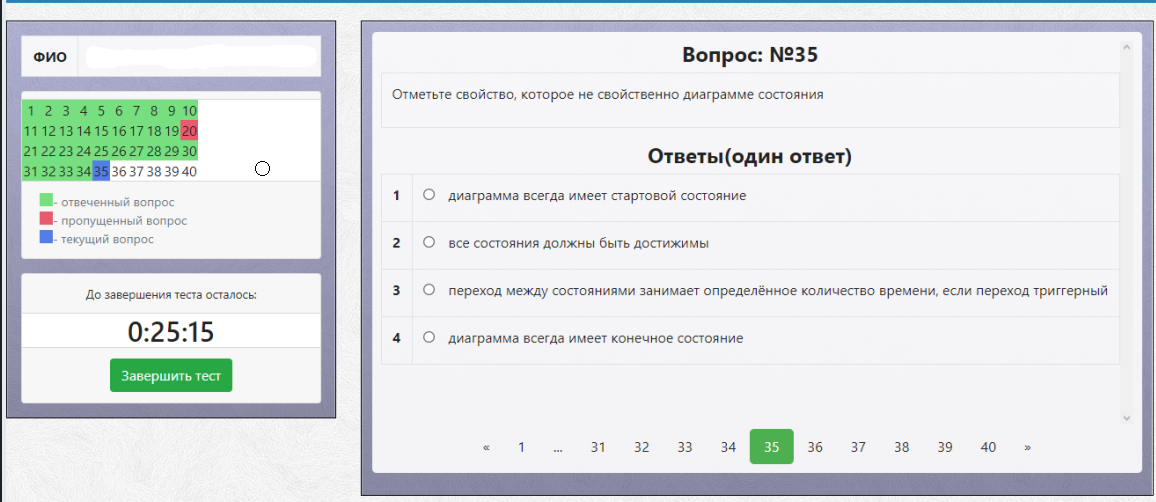


**Рис. 2 Тест на Google Form**

На рисунке изображено Тест сделанный Google Form. Goggle form предоставляет **много** вариантов создания теста. Нас интересует тест с несколькими вариантами ответов. Вопрос создаться легко, можно также выставить балл за вопрос и перемешать ответы, а также сделать вопрос обязательны, чтобы студент не пропустил его. Очень удобно. Однако, чтобы студент прошел тест, нужно либо отправить лично каждому студенту, либо отправить ссылку в группу. Что же до баллов, они известны только пользователю, который создал тест, преподаватель видит только количество сдавших и статистику ответов на вопрос. Преподавателю придется у каждого студента спросить сколько баллов они получили и потом самостоятельно внести баллы в ведомость. И нет ограничений по времени.

**AVN**

AVN – внедрение и сопровождение информационных систем для комплексной автоматизации бизнес-процессов учебных заведений, а также по созданию единой информационной системы управления образованием всей республики.



**Рис. 3 Тест в AVN - портале**

AVN – система которой пользуются на данный момент университет, в ней все разработана для образовательных учреждений. На картинке мы видим, как выглядит тест, тут есть и время и даже статус вопроса. Перед тестом идет авторизация студента, поэтому тест идёт на его профиле, баллы, полученные после теста, отправятся сразу в ведомость студента. Минусы, этой под программы, что каждого студента сотрудник деканата должен регистрировать на тест вручную, только после этого он появится в назначенное время. Из-за этого бывают сбои или опоздания теста, на странице студента.

**Вывод**

Google Form, имеет хорошую форму для составления теста, однако его невозможно интегрировать с образовательным сервисом университета поэтому выставление баллов, контролирование время, отправление теста и просматривание баллов и т.д. делает использование приложения затруднительным, большинство действий придется делать вручную. Это затормозит роботу университета и постоянно будут какие-нибудь инциденты.

Второй аналог – avn, нет особых минусов, так программа много лет используется образовательным учреждением университета. Но ему требуются некоторые доработки, например автоматическая регистрация студента, поиск и вывод задолжников и т.п. Так как приложение уже много лет используется, его интерфейс устарел и его нужно обновить и добавить функционал, который еще больше улучшит программу.

Наша программа — это улучшенный avn тестирование, в которой мы реализуем, как и функционал avn-тестирования, так и свой функционал. Это позволит еще больше, облегчит работу сотрудников университета и заметно увеличит скорость работы.

## **Проектирование**

* 1. Контекст моделирования

### Диаграмма бизнес-процессов (IDEF0)

**IDEF0** — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

Стандарт IDEF0 представляет организацию как набор модулей, здесь существует правило — наиболее важная функция находится в верхнем левом углу, кроме того, есть правило стороны:

* стрелка входа всегда приходит в левую кромку активности,
* стрелка управления — в верхнюю кромку,
* стрелка механизма — нижняя кромка,
* стрелка выхода — правая кромка.

При составление бухгалтерского баланса ориентируется на такие стандарты **управления** как:

* **Методические пособия –** материалы для составления теста.
* **Устав Университета и нормативные документы –** приказы, правила и нормативные документы, необходимые для работы.

**Входными данными** тут являются следующие элементы:

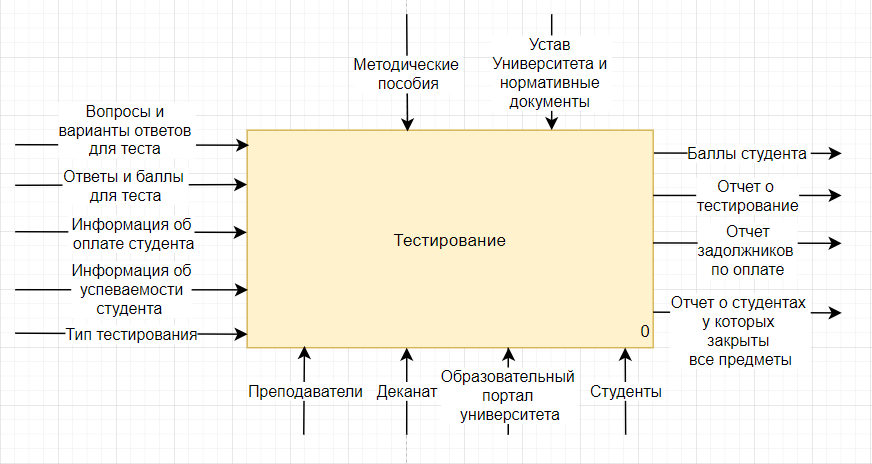
* **Вопросы и варианты ответов для теста –** вопросы для теста и варианты ответов для тестирования.
* **Ответы и баллы для теста –** ответ на вопрос и баллы за вопрос.
* **Информация об оплате студента –** информация о том, оплатил ли студент за учебу.
* **Информация об успеваемости студента –** электронная ведомость студента (есть ли задолженности по предметам).
* **Тип тестирования –** выбор из 3 видов тестирования.

**Механизмом** данного предприятия является:

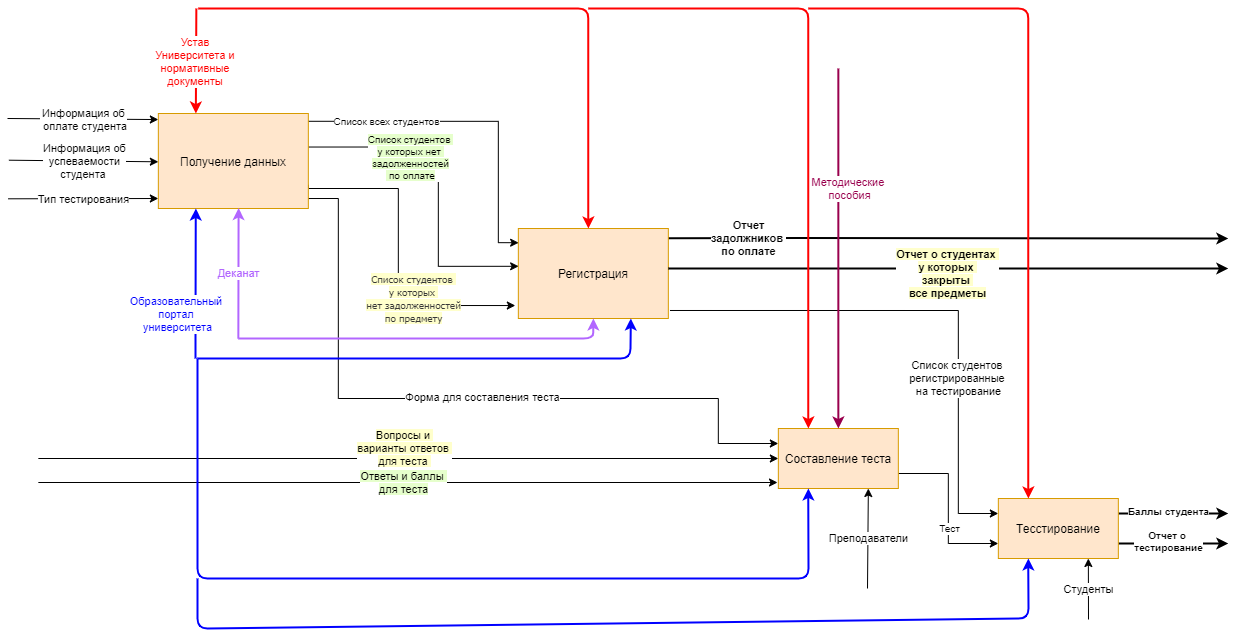
* **Преподаватели** – преподаватели университета
* **Деканат** – сотрудники университета
* **Образовательный портал Университета** – общий портал университета
* **Студенты** – учащиеся в университете

**Выходные данные** данного предприятия является:

* **Баллы студента –** баллы за тестирование
* **Отчет о тестирование –** отчет о том, как сдали студенты тестирования
* **Отчет задолжников по оплате –** список студентов, которые оплатили за учёбу
* **Отчет о студентах, у которых закрыты все предметы –** список студентов, у которых закрыты **(**история, география, кыргызский язык 1/2**)**



**Рис. 4 IDEF0 (Модель AS-IS в виде контекстной диаграммы тестирование)**



**Рис. 5 Декомпозиция IDEF0**

Диаграмма IDEF0 в контексте Тестирование, необходимо для упрощенного понимания, того, как происходит процесс Тестирования, в диаграмме были использованы элементы управления, механизм, а также входные и выходные данные, где каждый элемент тесно связан с друг-другом, итогом стало описание каждого бизнес-процесса, в данной программе.

### Диаграмма потоков данных (DFD)

**DFD** — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

**Диаграмма потоков данных** (data flow diagram, DFD) — один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML. Несмотря на имеющее место в современных условиях смещение акцентов от структурного к объектно-ориентированному подходу к анализу и проектированию систем, «старинные» структурные нотации по-прежнему широко и эффективно используются как в бизнес-анализе, так и в анализе информационных систем.

Выполняется декомпозиция в формате **DFD** функции(блока) прием товара.

В данном блоке будет описание сущностей, работ, и однонаправленных стрелок, созданных в данной функции.

**Сущностями** будут являться:

**Студент –** сущность, который сдаёт тестирование

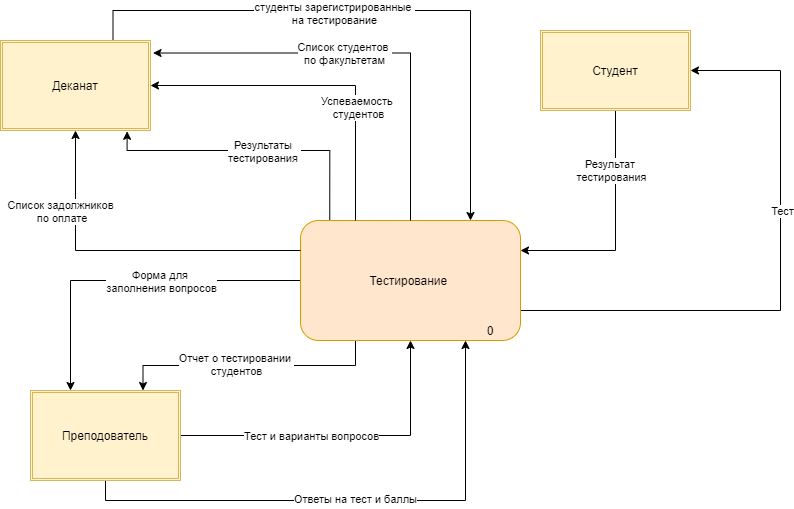
**Деканат –** сущность, которая регистрирует студентов на тестирование

**Преподаватель –** сущность, которая составляет тест

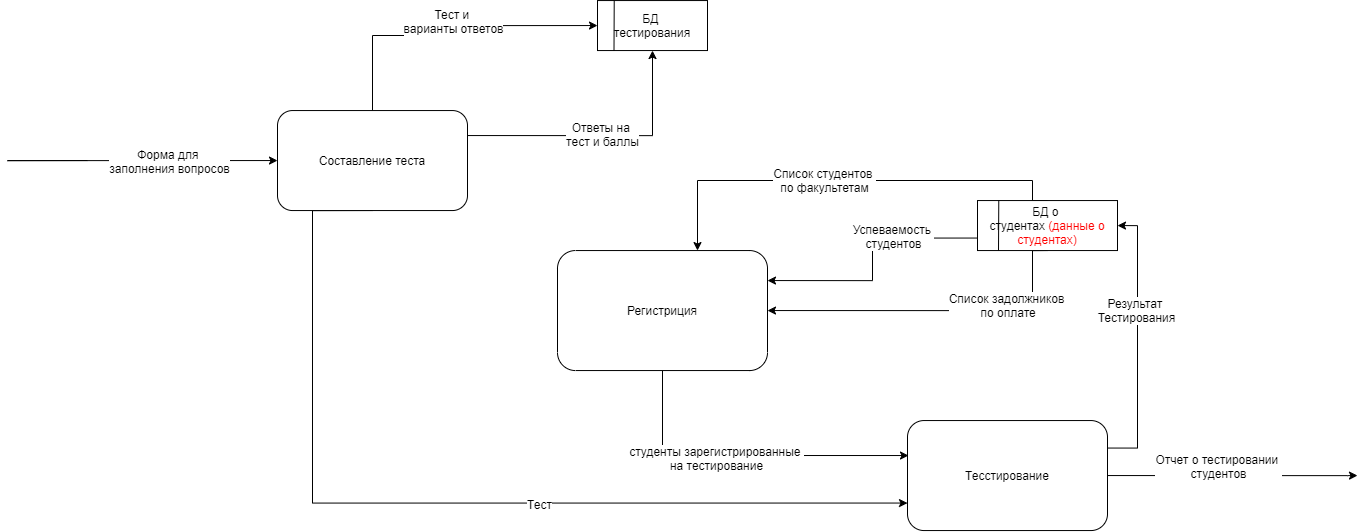
**Хранилищами данных** будут являться:

**БД тестирования** - в данном хранилище будут храниться данные о тесте (вопрос, варианты вопросов, баллы за вопрос)

**БД студента** - в данном хранилище будут храниться данные о студенте (успеваемость, оплата и тп)



**Рис. 6 DFD (Процесс передачи данных Тестирования)**

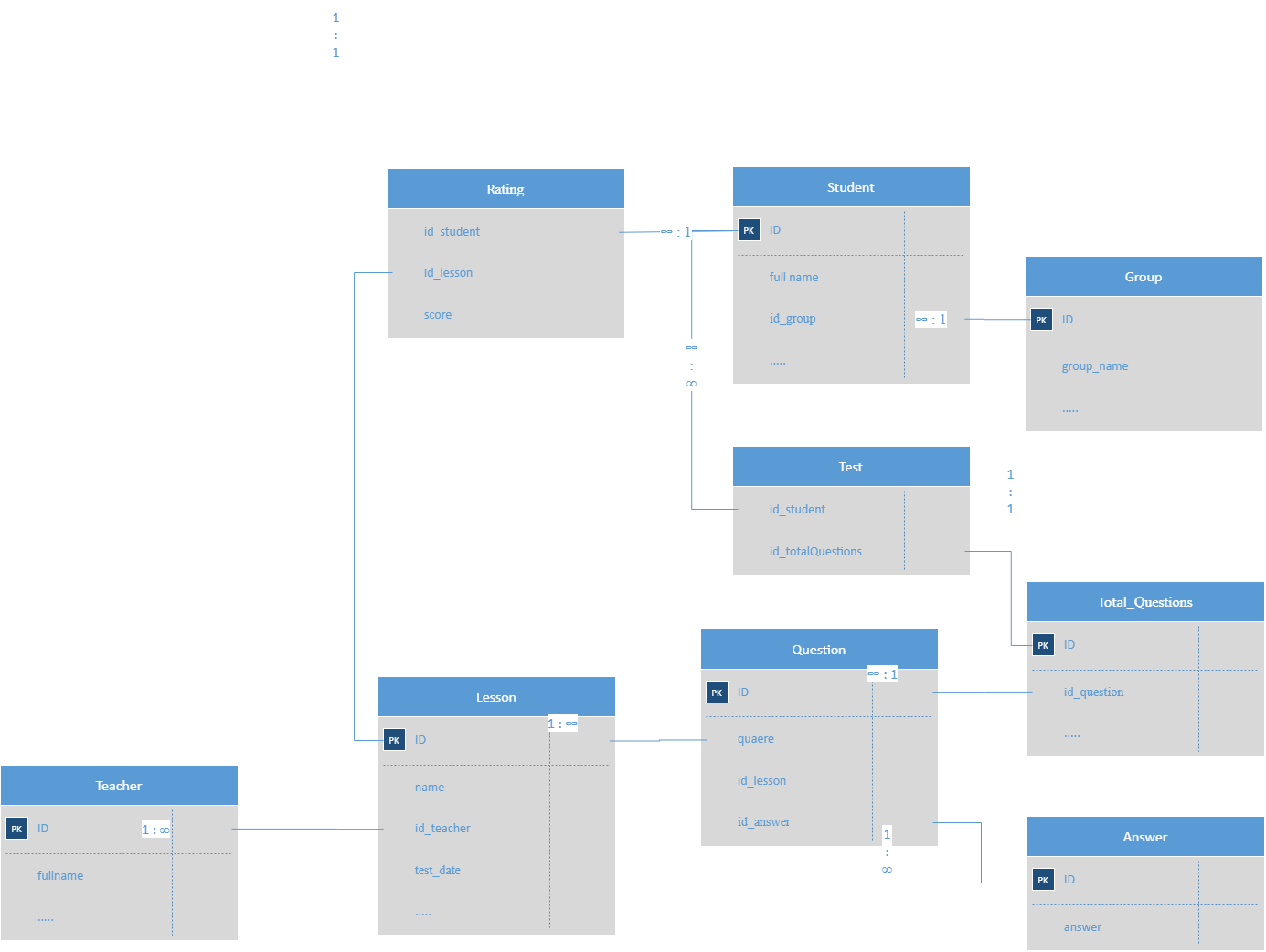
****

**Рис. 7 Декомпозиция DFD (Расписан процесс Тестирования и связь его с БД)**

* 1. Диаграмма ERD (ER-Модель)

ER-модель (от англ. entity-relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при концептуальном проектировании баз данных. С ее помощью можно выделить сущности и обозначить связи, которые могут устанавливаться между этими сущностями. Во время проектирования баз данных происходит преобразование ER-модели, описывающей представление данных на логическом уровне, в конкретную схему базы данных (физическую модель).

Для под программы Тестирование была разработана структура данных ER-модель, представлена на рис.4, которая является каркасом при построении БД программы

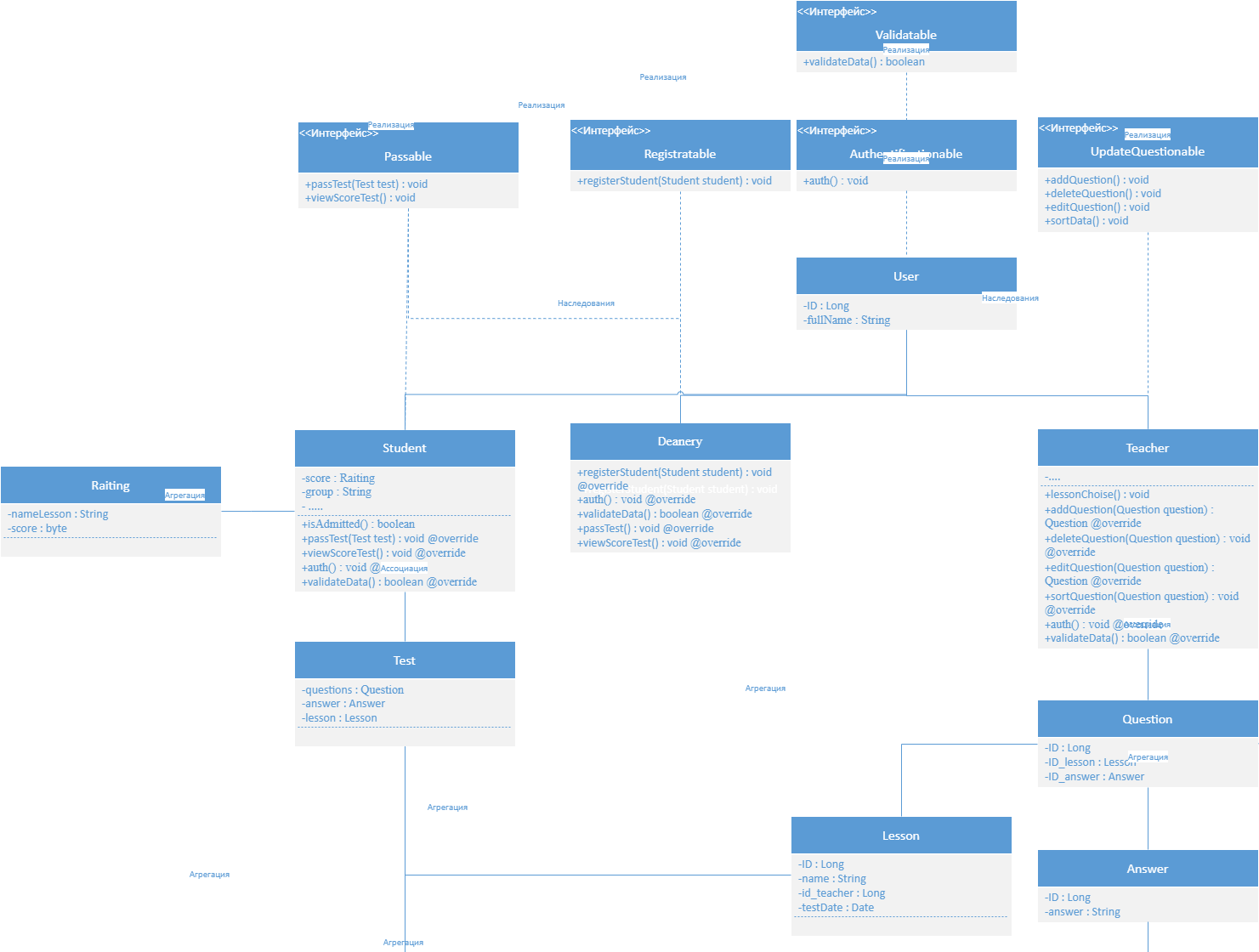


**Рис. 8 ERD- диаграмма**

* 1. Диаграмма классов (Сlass diagram)

Диаграмма классов — это статическое представление, используемое в области программирования для описания структуры системы, представления ее классов, атрибутов, операций и отношений между объектами.

На **рис.9** представлена диаграмма классов под программы Тестирования, в нем описаны классы и его методы.



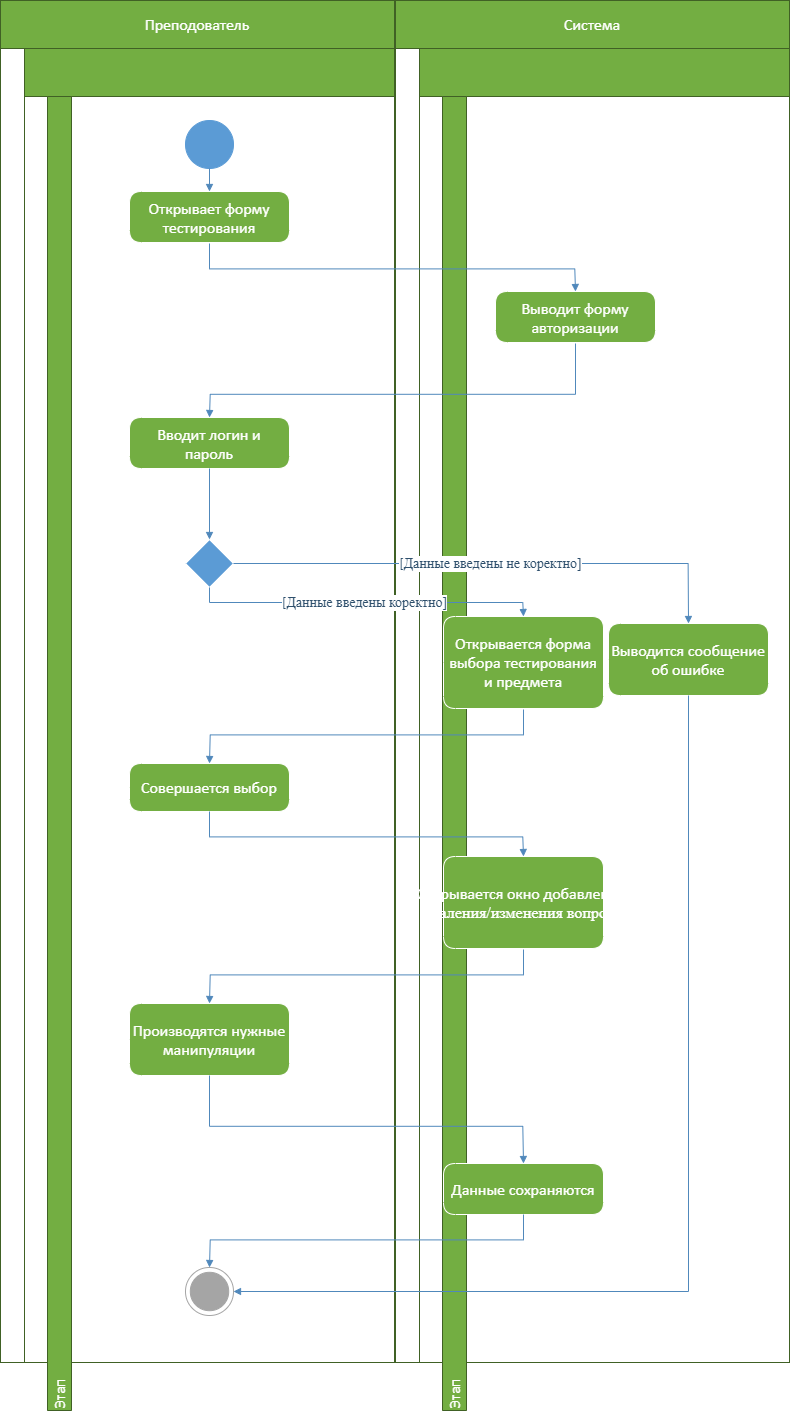
**Рис. 9 Диаграмма классов**

* 1. Диаграммы деятельности (Activity diagram)

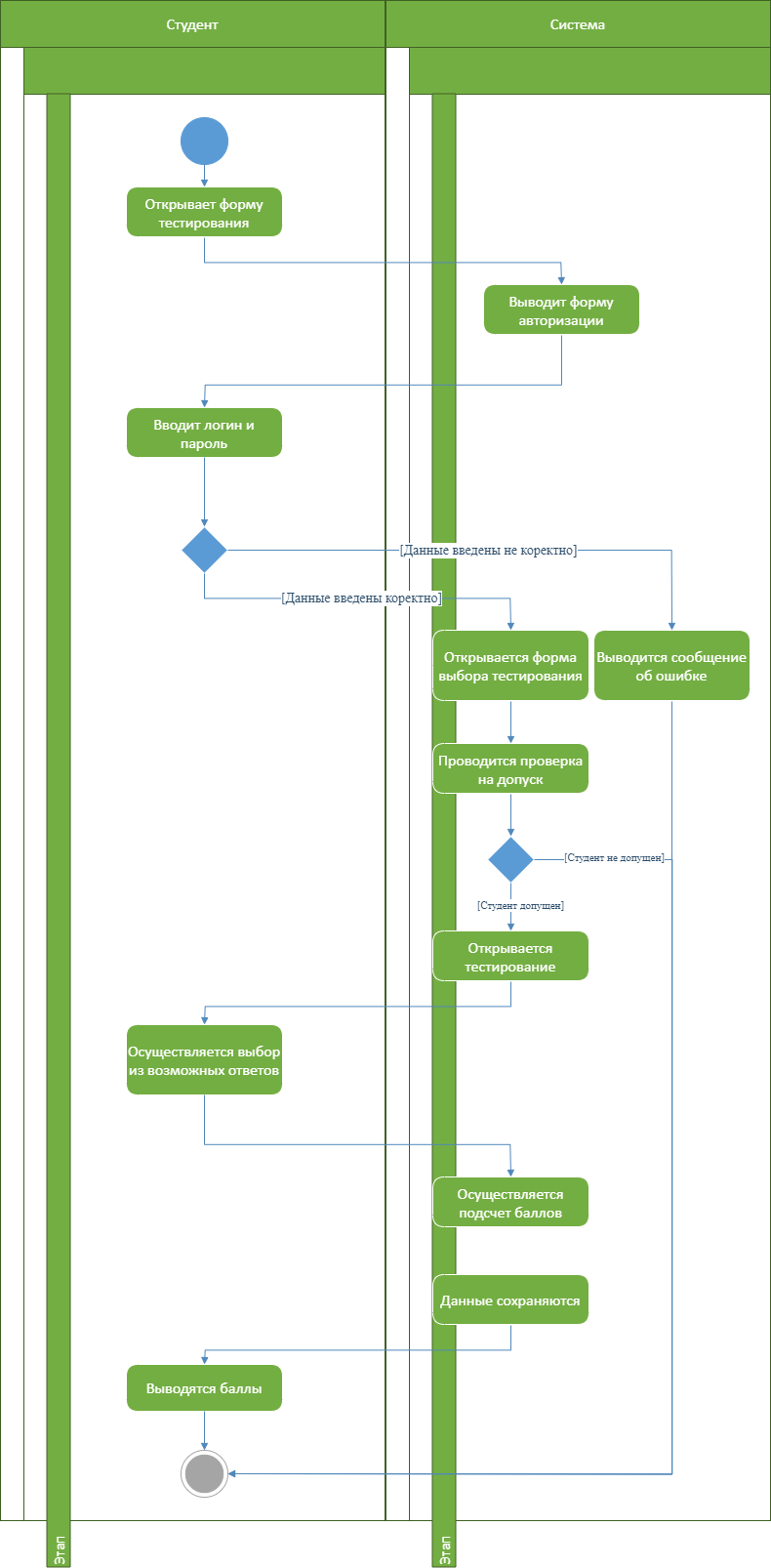
**Диагра́мма де́ятельности** (англ. activity diagram) — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью (англ. activity) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий англ. action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

На основе вариантов использования и их спецификаций были созданы диаграммы деятельности, отражающие работы функций системы.



**Рис. 10 Activity описывающий составления теста преподавателем**



**Рис. 11 Activity описывающий сдача Тестирования**

* 1. Диаграмма последовательности (Sequence diagram)

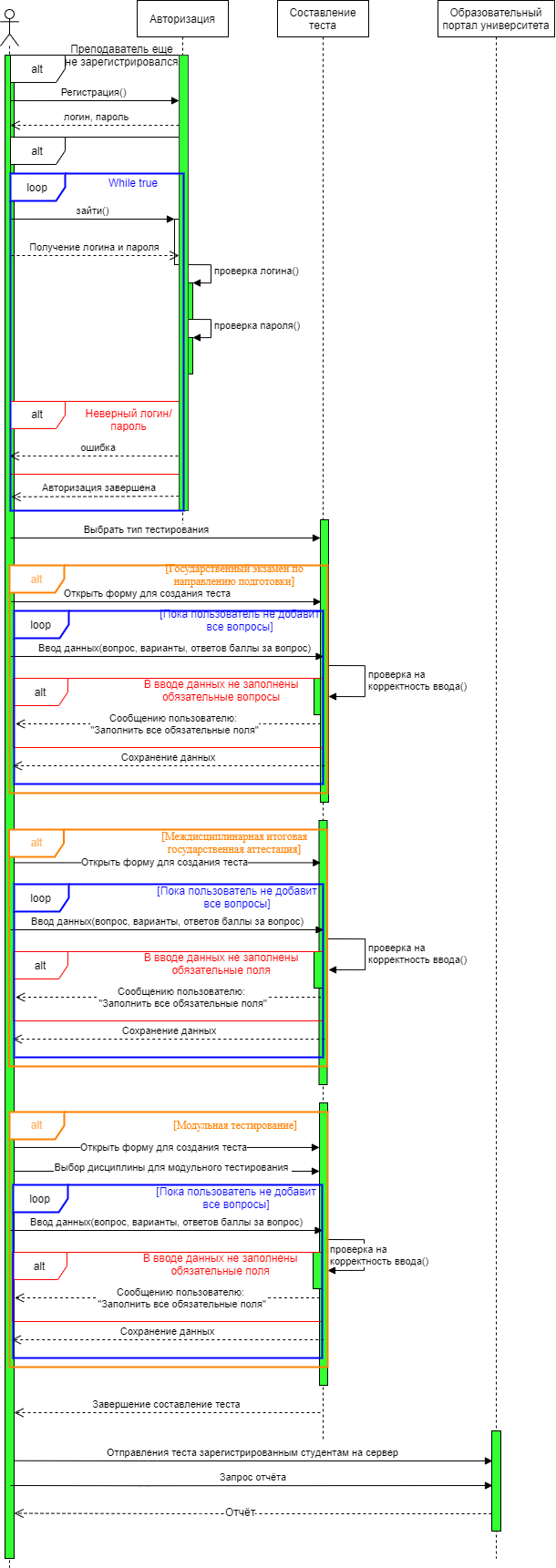
**Диаграмма последовательности** (англ. sequence diagram) — диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов).

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (англ. lifeline), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение объектом определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

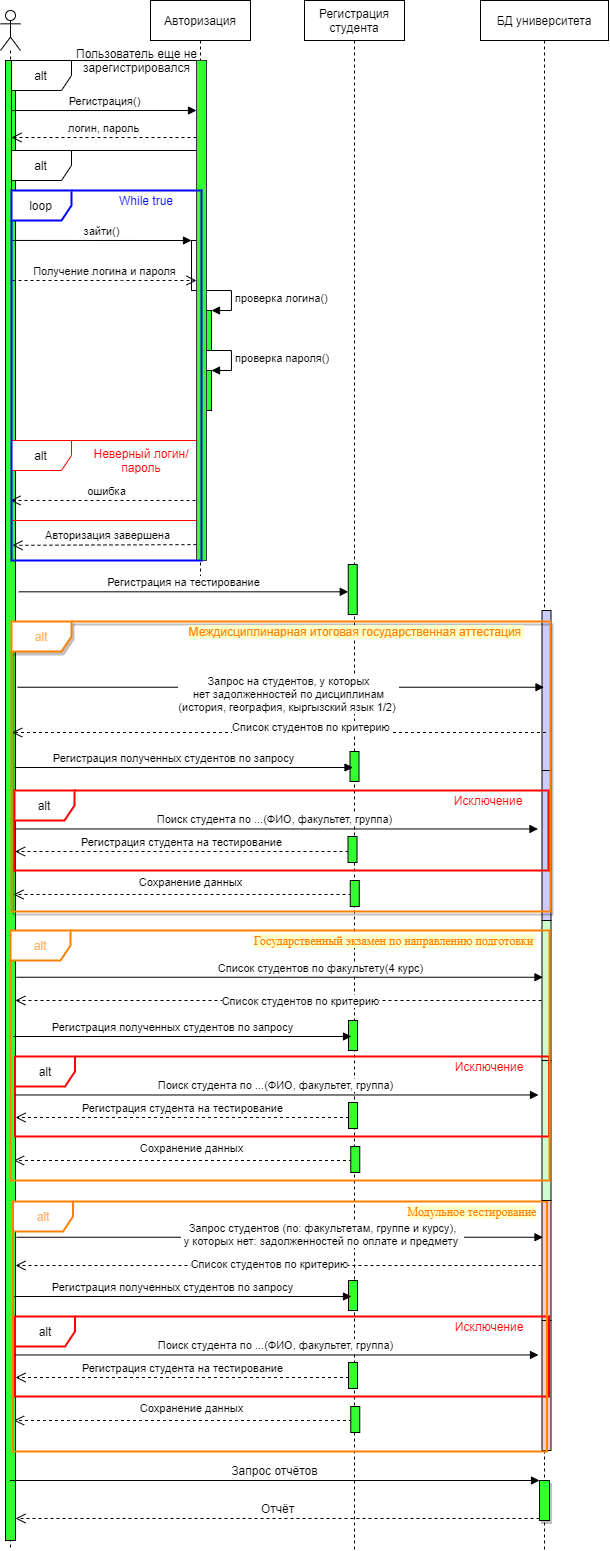
На **рис. 12** предоставлена диаграмма последовательности, в ней описываются Авторизация пользователя и составление теста преподавателем.

На **рис. 13** предоставлена диаграмма последовательности, в ней описывается Авторизация пользователя и регистрация студента

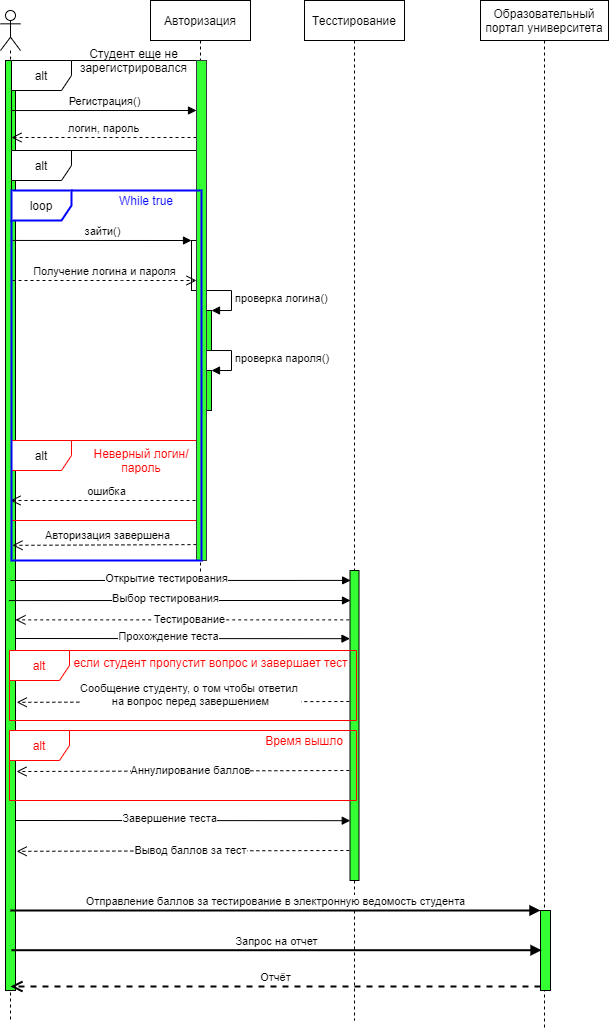
На **рис. 14** предоставлена диаграмма последовательности, в ней описывается Авторизация пользователя и прохождение теста студентом



**Рис. 12 Диаграмма последовательности (Авторизация + составление теста)**



**Рис. 13 Диаграмма последовательности (Авторизация + Регистрация студента на тестирование)**



**Рис. 14 Диаграмма последовательности (Авторизация + Тестирование)**

## **Заключение**

Итогом проделанной работы стало закрепление навыков, приобретенных за 3-й Курс специальности, Программная инженерия, и создание программы Тестирования, это послужит ценным опытом при создании и проектировании программного обеспечения.

Для создания программы были проведены конструкторские работы с созданием необходимых диаграмм бизнес-процессов и потоков данных (IDEF0, DFD), а также были использованы UML диаграммы для описания функций, перехода состояний, и сценариев взаимодействия программы с пользователем.

## **Литература**

* Современные наукоемкие технологии // Педагогические науки // ТЕСТИРОВАНИЕ КАК ФОРМА ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ – 2016. – 37.013 – № 7 (часть 1) – С. 118-121
* <https://avn.kstu.kg> – Образовательный портал университета
* <https://avn.kg/about.html>
* <https://netology.ru/blog/google-formy> - Все возможности Google Forms
* <https://kstu.kg/universitet/5-kolonka/normativnye-dokumenty/zagolovok-po-umolchaniju-4-1> - нормативные документы КГТУ