**1. ВВЕДЕНИЕ**

В данном курсовом проекте производится проектирование и реализация программной системы организации дистанционного прохождения системы тестов.

На данный момент, множество университетов и других образовательных учреждений проводят тесты в неинтерактивном режиме.

Программа предоставляет широкие возможности студентам и преподавателям. Пользуясь любым устройством, имеющим подключение к сети интернет, они могут проходить тесты по разным предметам, предусмотренные учебным процессом. Введение интерактивных тестов и лекций, помимо бумажных аналогов, позволит контролировать успеваемость, а также грамотно планировать выполнение домашней работы, контрольных и рейтинговых работ, что, в свою очередь, отразится на успешном выполнении учебного плана.

В ходе выполнения курсового проекта, будут решены следующие задачи:

- анализ предметной области;

- разработка проекта программной системы;

- разработка прототипа программной системы.

Результатом выполнения курсового проекта будет работающее программное обеспечение, позволяющее вести учет дистанционной образовательной деятельности учебного заведения.

В ходе выполнения курсового проекта выполнена разработка проекта информационной системы с использованием языка UML, реализация диаграмм с использованием среды Microsoft Office Visio, реализация программного обеспечения с использованием технологий разработки ASP .NET MVC, Entity Framework. Среда программирования – Microsoft Visual Studio 2015.

1. **Постановка задачи**

Необходимо разработать программное обеспечение, ведущее учет дистанционного обучения образовательного учреждения.

Для решения поставленной задачи, необходимо последовательно выполнить этапы анализа, проектирования и реализации системы.

Анализ предметной области. Результатом данного этапа работы является описание бизнес-процесса учета знаний студентов и словарь предметной области.

Проектирование системы. Для проектирования системы необходимо использовать язык UML и включать в себя следующие UML-диаграммы:

- диаграмма прецедентов;

- диаграмма классов;

- диаграмма состояний;

- диаграмма последовательностей.

Также этап проектирования включает схему базы данных в виде ER-диаграммы логической моделей данных.

Программная реализация.

Минимальный набор функций:

1. создание и поддержка справочников студентов и преподавателей;
2. создание и поддержка справочников тестов с возможностью включения студента в дистанционное прохождение;
3. прохождение тестов;
4. выставление оценок;

Система должна быть реализована на языке C# в виде веб-приложения с использованием технологий ASP .NET MVC.

1. **Описание предметной области**
   1. **Общие сведения о предмете автоматизации**

Темой данного курсового проекта является закрытая программная система учёта дистанционного обучения в образовательном учреждении. Подобная программная система позволяет образовательным учреждениям расширить функциональные возможности как лекторов, так и учеников. Пользователями системы дистанционного обучения станут студенты и преподаватели (роль «Студент» и «Преподаватель»), а также руководство учебного заведения. За развёртывание и поддержку работоспособности системы в рамках данной предметной области отвечает системный администратор.

Программа предоставляет широкие возможности студентам и преподавателям. Пользуясь любым устройством, имеющим подключение к сети интернет, они могут просматривать оценки, лекции, узнавать домашнее задание, получаемое в рамках учебного процесса. Ведение электронного дневника и журнала, помимо бумажных аналогов, позволит контролировать успеваемость, а также грамотно планировать расписание и выполнение домашней работы, что, в свою очередь, отразится на успешном выполнении учебного плана.

Пользуясь программной системой дистанционного обучения персонал так же получит широкие возможности для планирования и контроля учебного процесса. В системе будут реализованы следующие функции: выставление оценок, составления заданий и просмотр статистики. Возможность работы с системой, пользуясь любым устройством имеющем доступ к интернету, намного удобней, чем заполнение бумажного аналога в рамках одного занятия. Сформированная программной системой статистика позволит преподавателю наглядно представить насколько хорошо студенты группы усвоили какой-либо материал.

Администрация образовательного учреждения с помощью программной системы сможет добавлять преподавателей и студентов в базу данных, создавать списки новых и текущих групп и формировать для них удобное расписание. Система предусматривает три роли для администрации школы: секретарь, диспетчер и декан (факультета или университета в целом). При этом происходит разделения обязанностей. Функции добавления студентов, формирования групп и расписания будут доступны секретарю, при этом декан получит возможность добавления персонала образовательного учреждения, а диспетчер возьмёт на себя функции составления расписания.

Системный администратор занимается развёртыванием и поддержкой программной системы в рамках какого-либо образовательного учреждения. Данный работник имеет все основные функции системы: добавление/редактирование/удаление списка пользователей, просмотр/изменения/удаление записей в справочниках, формирование/удаление групп и материалов – что позволит администратору поддерживать актуальность справочников студентов и преподавателей, а также контролировать правильность заполнения справочников.

Регистрация новых пользователей производится персоналом учебного заведения.

* 1. **Словарь предметной области**

Программная система - программная система дистанционного обучения студента в образовательном учреждении.

Гость – любой неавторизированный участник программной системы. Не имеет доступа к каким-либо функциям.

Пользователь – какой-либо зарегистрированный участник программной системы. Каждый пользователь имеет единственную определенную роль.

Справочник – фрагмент базы данных, хранящий информацию о группе определенных объектов. Определенные группы пользователей могут выполнять редактирование/добавление этих справочников, а также информации в них.

Группа – совокупность студентов, выделенных в отдельную группу. Каждая группа имеет свой уникальный перечень предметов, тестов и домашних заданий.

Электронный дневник – предоставляет возможность просмотра оценок и домашнего задания студента.

Электронный журнал – предоставляет функции просмотра справочника групп и студентов.

Студент – роль при которой пользователь, который имеет функцию просмотра своего электронного дневника, а также принадлежит к какой-либо группе. Регистрацию в системе производит заведующий учебной частью или директор.

Преподаватель – роль, при которой пользователь имеет доступ к электронному журналу. Регистрацию в системе производит директор.

Декан – роль, при которой пользователь может подтверждать внесённые изменения и управлять создание пользователей и предметов. Учетная запись уникальна и заведомо добавлена в базу данных программной системы.

Персонал – пользователь, который имеет одну из следующих ролей: преподаватель, секретарь, диспетчер, декан.

Расписание– расписание предметов для групп студентов. Так как форма обучения дистанционная, данная сущность содержать информацию о сроках предоставления работ.

* 1. **Сценарий взаимодействия пользователя с системой**

В приложении используется разделение по ролям. В соответствии с должностями, были получены следующие роли:

* Студент
* Преподаватель
* Декан
* Диспетчер
* Секретарь

Пользователь с ролью Студент может:

* Просматривать свои оценки, лекции, предметы и задания по ним, согласно своей группе
* Просматривать информацию о предмете или тесте
* Проходить тесты
* Смотреть лекции

Пользователь с ролью Преподаватель может:

* Просматривать список групп и студентов;
* Просматривать список предметов;
* Выставлять оценки студентам;
* Добавлять предметы, домашние задания по ним и тесты группе.

Пользователь с ролью Секретарь может:

* Создавать пользователей, добавлять их в группу и редактировать информацию о них

Пользователь с ролью Диспетчер может:

* Создавать расписание, отправлять его на проверку декану и редактировать.

Пользователь с ролью Диспетчер может:

* Создавать и изменять учебный план и отправлять его на проверку декану;

Пользователь с ролью Директор может также:

* Подтверждать созданный учебный план.
* Создавать группы и помещать в них студентов.
* Просматривать список пользователей ПС, а также осуществлять их добавление.

Гость имеет возможность:

* авторизоваться;
* зарегистрироваться;

1. **проектирование системы**
   1. **Диаграмма прецедентов**

Объект информатизации исследуется в границах одного образовательного учреждения. Действующими лицами (в нотации UML – актерами) являются: Студент, Преподаватель, Диспетчер, Секретарь и Гость. Ниже приведен перечень и описание некоторых прецедентов в проектируемой системе.

***Название:*** «Просмотр расписания»

***Предусловие:*** пользователь имеет доступ к веб-сайту программной системы.

***Действующее лицо:*** студент

***Основной поток:*** Действующее лицо открывает окно приложения, отображающую информацию, заполненную в дневнике. С помощью клавиш навигации осуществляет выборку необходимой информации. Прецедент при этом использует функции просмотра оценок и просмотра заданий. При завершении работы действующее лицо закрывает окно приложения.

***Название:*** «Управление учебным планом»

***Предусловие:*** персонал имеет доступ к веб-сайту программной системы.

***Действующее лица:*** диспетчер.

***Основной поток:*** действующее лицо открывает форму управления расписанием и выполняет необходимые ему действия с помощью выпадающих списков, также с помощью клавиш навигации имеет возможность активировать прецедент «редактирование расписания». В окне представлено сформированное расписание для выбранной группы и кнопки их редактирования. При завершении работы действующее лицо сохраняет учебный план и закрывает окно приложения.

***Название:*** «Добавить пользователя»

***Действующее лица:*** секретарь.

***Основной поток:*** действующее лицо открывает форму заполнения информации о пользователе. Выполняет корректный ввод данных в формы (выполняет ввод следующих полей: email, логин, пароль) выбирает роль из выпадающего списка и нажимает кнопку «Добавить». Система создает нового пользователя с выбранной ролью и сохраняет данные в БД.

Альтернативный поток:

* Действующее лицо неправильно заполнило поля, при этом получило сообщение об ошибке. В таком случае, данный пользователь должен перепроверить и исправить введенные данные.
* Действующее лицо не желает добавлять нового пользователя по каким-либо причинам и нажимает кнопку «Отмена». Система не создает нового пользователя и не сохраняет введенные данные, происходит переадресация на предыдущую страницу.
* Если введенные ключевые поля принадлежат уже зарегистрированному пользователю, система выдает сообщение об ошибке.

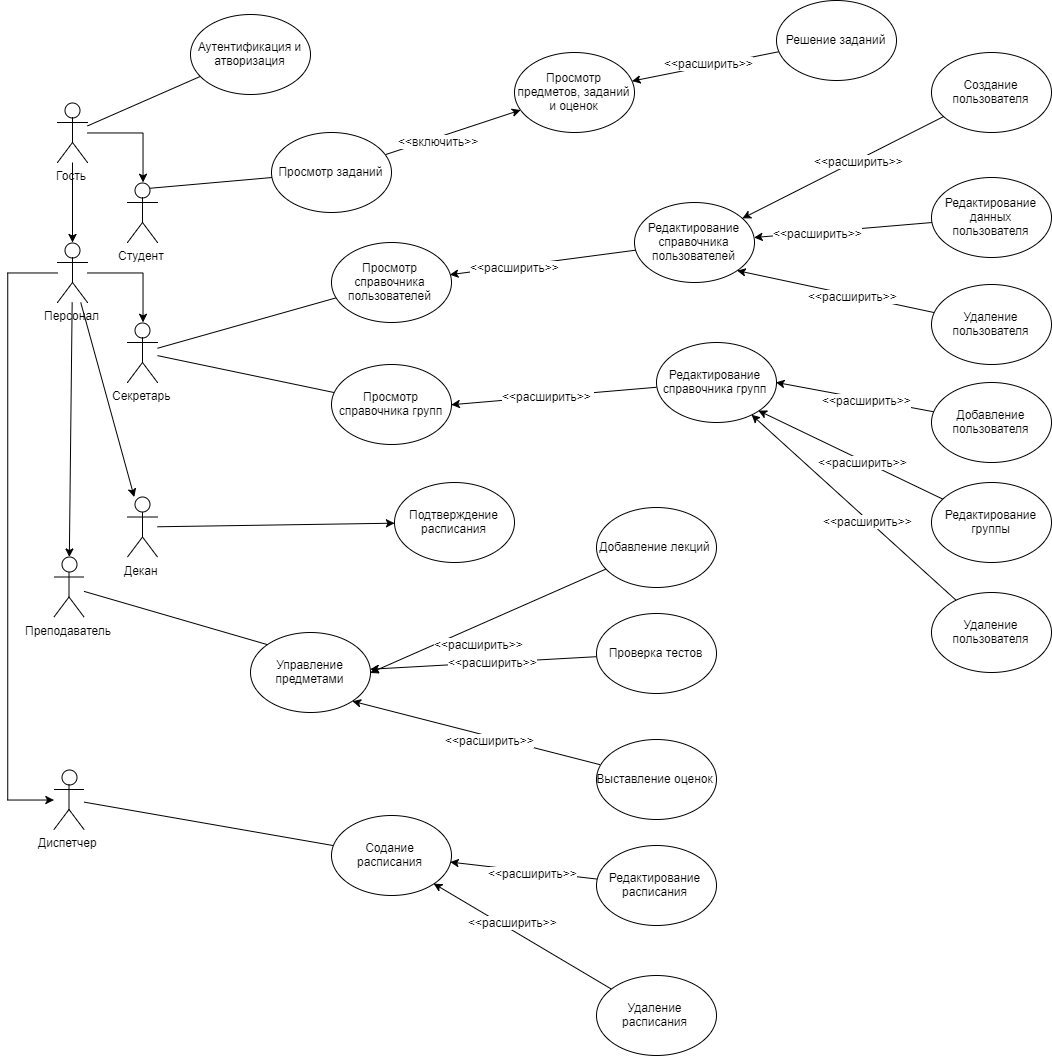
Диаграмма прецедентов представлена на рисунке 1.

Рисунок 1. Диаграмма прецедентов

* 1. **Диаграмма классов.**

Для проектирования структур и моделей данных программной системы, была разработана диаграмма классов, представляющая основные сущности системы. Диаграмма классов представлена на рисунке 2.

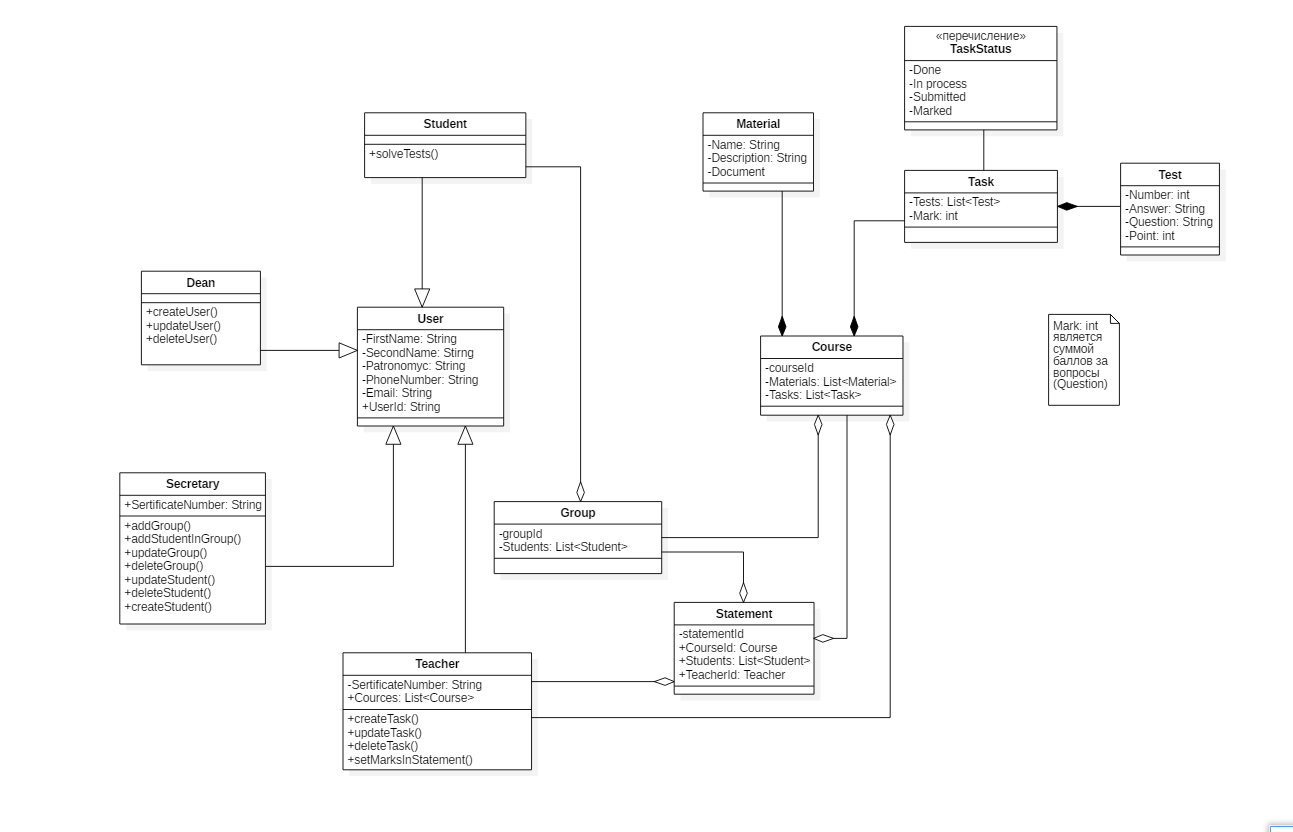


Рисунок 2. Диаграмма классов

В результате анализа были выделены следующие основные сущности:

*Group (Группа) –* сущность, объединяющая студентов и преподавателей. Группа имеет свой определённый идентификатор. Предполагается, что студент обязательно прикреплён к какой-то группе. И совокупность студентов, изучающих одни и те же предметы, называются группой*.*

*Schedule (Расписание) – расписание предметов для групп студентов. Так как форма обучения дистанционная, данная сущность содержать информацию о сроках предоставления работ.*

*Task (Задание) –* материал, содержащий упражнения или тестовые задания, которые необходимо выполнить студенту. Создаётся преподавателем (Teacher).

*TaskStatus (Статус выполнения задания)* – отображает этап выполнения задания. Может принимать такие значения:

* *Done (Выполнено)* – студент выполнил задание и ожидает оценки. Если данное значение истинно, то работа может быть оценена, иначе – не может быть оценена.
* *In process (В процессе)* – студент получил домашние задание и в данный период времени выполняет его.
* *Mark* (Оценка) – преподаватель оценил выполненное домашнее задание.

*Mark (Оценка)* – число, оценивающее работу студента по определённому предмету;

*Subject (Предмет) –* отображает название предмета;

*Student(Студент)* – пользователь программной системы, использующий её для дистанционного обучения;

*Teacher(Преподаватель)* – пользователь программной системы, который выполняет функции проверки и загрузки домашнего задания;

*Dispatcher (Диспетчер)* – пользователь программной системы, который выполняет функции создания и загрузки расписания.

*Secretary (Секретарь)* – пользователь программной системы, который принимает расписание и управляет всеми пользователями.

* 1. **Диаграмма состояний.**

Важнейшим объектом ПС являются задания.

Состояние объекта «Задание».

1. Пользователь (преподаватель) составляет тесты и добавляет их. Объект «Задание» создан в системе. Его статус – «В процессе».
2. Пользователь (студент) приступает к выполнению заданий. Ответив на все вопросы, он отправляет задание на проверку. Оно приобретает статус «Представлено на проверку».
3. После отправки преподаватель должен проверить правильность выполненных заданий. Если его удовлетворяет работа студента, он оценивает задание. Оно переходит в статус «Оценено на (какую-то оценку)».
4. Если студента не удовлетворяет оценка, он может выполнить его заново.

Диаграмма состояний представлена на рис.4.

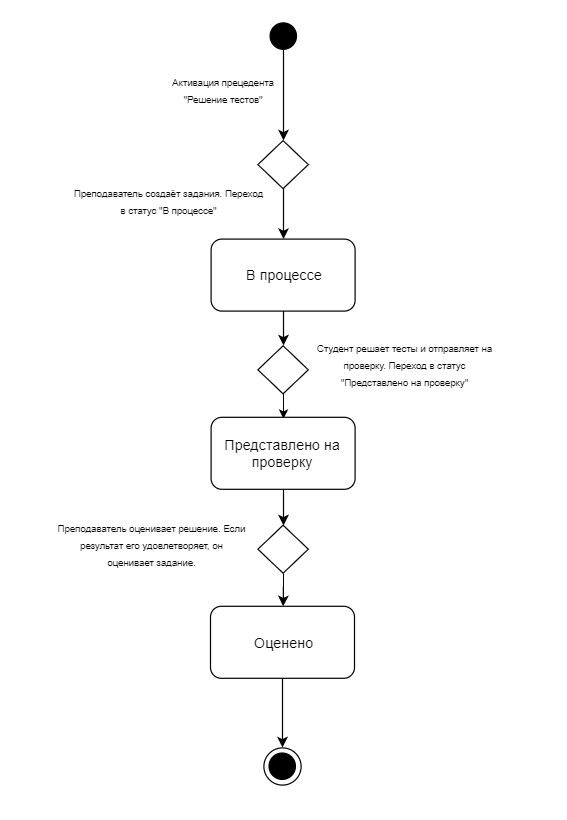


Рисунок 4. Диаграмма состояний объекта «Задание»