ДИПЛОМЕН ПРОЕКТ ЗА ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ

по професия код 481030 „Приложен програмист“

специалност код 4810301 Приложно програмиране“

Тема: „Платформа за електронно обучение“

Автор:

Борис Бисеров Канев, клас XI В

Ръководител:

Антони Димитров

БургасСЪДЪРЖАНИЕ

[1 Увод 3](#_Toc119855807)

[2 Цели и обхват на софтуерното приложение 3](#_Toc119855808)

[3 Анализ на решението 3](#_Toc119855809)

[3.1 Потребителски изисквания и работен процес 3](#_Toc119855810)

[3.2 Примерен потребителски интерфейс 5](#_Toc119855811)

[3.3 Диаграми на анализа 5](#_Toc119855812)

[3.4 Модел на съдържанието / данните 5](#_Toc119855813)

[4 Дизайн 6](#_Toc119855814)

[4.1 Реализация на архитектурата на приложението 6](#_Toc119855815)

[4.2 Описание на слоевете, предназначението им, библиотеки и методи включени в съответния слой. 6](#_Toc119855816)

[4.3 Организация и код на заявките към база от данни 6](#_Toc119855817)

[4.4 Наличие на потребителски интерфейс (конзолен, графичен, уеб) 6](#_Toc119855818)

[5 Ефективност и бързодействие на решението 6](#_Toc119855819)

[6 Тестване 6](#_Toc119855820)

[7 Заключение и възможно бъдещо развитие 6](#_Toc119855821)

[8 Използвани литературни източници и Уеб сайтове 7](#_Toc119855822)

[9 Приложения 7](#_Toc119855823)

[10 Критерии и показатели за оценяване 8](#_Toc119855824)

# Увод

Проектът е цялостна система, предназначена да улесни управлението на тестове, задачи и уроци за образователните институции. Системата е проектирана да предоставя лесна за използване платформа както за преподаватели, така и за ученици, с функции като вземане на тестове, изпращане на задачи и достъп до уроци. Системата е изградена с помощта на рамката ASP.NET и използва база данни за съхраняване на информация за тестове, въпроси и резултати.

Основната цел на проекта е да осигури гъвкава и мащабируема платформа за образователните институции, за да управляват своите процеси на тестване и задания. Системата е проектирана да бъде лесна за използване, с удобен за потребителя интерфейс, който позволява на преподавателите да създават и управляват тестове и задачи и позволява на учениците да имат достъп и да ги изпълняват.

Една от основните характеристики на проекта е функционалността за вземане на тестове. Преподавателите могат да създават тестове, използвайки различни типове въпроси, включително въпроси с избираем отговор, кратък отговор и въпроси за есе. Системата предоставя интерфейс за преподавателите за създаване на въпроси и присвояване на точкови стойности, както и възможност за персонализиране на инструкции за теста и критерии за оценяване. След това учениците могат да вземат тестовете онлайн, като техните резултати се записват в базата данни за по-късен преглед.

Друга ключова характеристика на проекта е функционалността за изпращане на задачи. Преподавателите могат да създават задачи, използвайки различни формати, включително текстови задачи, мултимедийни задачи и задачи по програмиране. Системата предоставя интерфейс за преподавателите за създаване на задания и определяне на крайни срокове, както и възможност за оценяване и предоставяне на обратна връзка за изпратените задания. След това студентите могат да изпращат задачите си онлайн, като техните предложения се записват в базата данни за по-късен преглед.

В допълнение към полагането на тестове и подаването на задачи, проектът също така предоставя достъп до уроци и материали за курса. Преподавателите могат да качват материали за уроци, включително видеоклипове, документи и друго мултимедийно съдържание, а студентите имат достъп до тези материали онлайн като част от тяхната курсова работа.

За да улесни управлението на тестове, задачи и уроци, проектът използва база данни за съхраняване на информация за тези компоненти. Базата данни включва таблици за тестове, въпроси, резултати, задачи, изпращания и друга подходяща информация. Тези данни се използват от системата, за да осигурят удобен за потребителя интерфейс за преподаватели и студенти, както и за проследяване и анализ на ефективността на потребителите.

Едно от основните предимства на проекта е неговата гъвкавост и мащабируемост. Системата е проектирана да бъде лесно персонализирана, с функции, които могат да се добавят или премахват в зависимост от нуждите на учебното заведение. Освен това системата може да обработва голям брой потребители и големи количества данни, което я прави подходяща за използване в различни образователни контексти.

Друго предимство на проекта е лекотата на използване. Системата е проектирана да бъде интуитивна и лесна за използване, с опростен и рационализиран интерфейс, който улеснява навигацията от преподаватели и ученици. Системата е достъпна и от различни устройства, включително настолни компютри, лаптопи, таблети и смартфони, което позволява на потребителите да имат достъп до системата от всяко място с интернет връзка.

Като цяло проектът е цялостна система, предназначена да улесни управлението на тестове, задачи и уроци за образователните институции. С функции като вземане на тестове, изпращане на задачи и достъп до уроци, системата предоставя лесна за използване платформа както за преподаватели, така и за ученици. Чрез използване на база данни за съхраняване на информация за тестове, въпроси и резултати, системата предоставя удобен за потребителя интерфейс за преподаватели и студенти, както и възможност за проследяване и анализиране на работата на потребителите. Със своята гъвкавост, мащабируемост и лекота на използване, проектът е идеално решение за образователни институции, които искат да рационализират своите процеси на тестване и възлагане

# Цели и обхват на софтуерното приложение

## Цели на софтуерното приложение

Проектът е разработен, за да отговори на редица цели, всички от които са предназначени да улеснят управлението на тестове, задачи и уроци за образователните институции. Тези цели включват:

1. Рационализиране на процесите на тестване и задание: Една от основните цели на проекта за училищна система е да рационализира процесите на тестване и задание за образователните институции. Системата предоставя лесна за използване платформа както за преподаватели, така и за ученици, с функции като полагане на тестове, изпращане на задачи и достъп до уроци. Използвайки системата, преподавателите могат да създават и управляват тестове и задачи по-ефективно, докато студентите могат да завършват курсовата си работа онлайн, без да е необходимо да подават документи на хартиен носител.
2. Подобряване на ангажираността на учениците: Друга цел на проекта за училищна система е да подобри ангажираността на учениците с тяхната курсова работа. Чрез осигуряване на достъп до уроци и материали за курсове онлайн, учениците могат по-лесно да преглеждат и изучават съдържанието, от което се нуждаят, за да успеят. Освен това системата предоставя на учениците интерактивна и ангажираща платформа за полагане на тестове и изпращане на задачи, което може да им помогне да ги мотивирате и вдъхновите да постигнат най-добрите си резултати.
3. Подобряване на сътрудничеството и комуникацията: Проектът служи и за подобряване на сътрудничеството и комуникацията между преподаватели и ученици. Системата предоставя платформа за преподавателите за предоставяне на обратна връзка и оценяване на задания и тестове, докато студентите могат да задават въпроси и да търсят разяснения относно курсовата работа, ако е необходимо. Това може да помогне за създаването на по-съвместна и подкрепяща учебна среда, което може да бъде особено ценно за студенти, които може да изпитват трудности с курсовата работа.
4. Проследяване и анализиране на ефективността на потребителите: Проектът също служи за проследяване и анализиране на ефективността на потребителите, което може да бъде ценно за преподаватели и администратори. Използвайки системата, преподавателите могат да наблюдават напредъка на учениците, да проследяват резултатите от тестовете и задачите и да идентифицират области, в които учениците може да изпитват трудности. Тези данни могат да се използват за информиране на решенията за преподаване и учебни програми, както и за предоставяне на ценна обратна връзка на учениците относно тяхното представяне.
5. Осигуряване на гъвкавост и мащабируемост: И накрая, проектът за училищна система служи за осигуряване на гъвкавост и мащабируемост за образователните институции. Системата е проектирана така, че да може лесно да се персонализира, с функции, които могат да се добавят или премахват в зависимост от нуждите на институцията. Освен това системата може да обработва голям брой потребители и големи количества данни, което я прави подходяща за използване в различни образователни контексти.

Като цяло проектът е разработен, за да отговори на редица важни цели, всички от които са предназначени да улеснят управлението на тестове, задачи и уроци за образователните институции. Чрез рационализиране на процесите, подобряване на ангажираността на учениците, подобряване на сътрудничеството и комуникацията, проследяване и анализиране на ефективността на потребителите и осигуряване на гъвкавост и мащабируемост, системата може да предостави ценен ресурс както за преподаватели, така и за студенти.

## Обхват на софтуерното приложение

Проектът е цялостно софтуерно приложение, предназначено да улесни управлението на тестове, задачи и уроци за образователни институции. Обхватът на приложението е широкообхватен, включващ разнообразие от характеристики и функции, които са предназначени да отговорят на нуждите както на преподаватели, така и на ученици.

Основните характеристики на проекта включват полагане на тестове, изпращане на задачи и достъп до уроци. Преподавателите могат да създават и управляват тестове и задания, като използват разнообразни формати на въпроси и задания, включително възможности за избор, кратък отговор, въпроси за есе и задачи по програмиране. Учениците имат достъп до тестовете и задачите онлайн, да ги попълнят и да ги изпратят за оценка и обратна връзка.

В допълнение към тези основни функции, проектът включва различни други функции и характеристики, като например:

1. Потребителски акаунти: Системата включва система за управление на потребителски акаунти, където преподаватели и ученици могат да създават и управляват своите акаунти. Това позволява на потребителите да имат защитен достъп до системата с подходящи разрешения.
2. Крайни срокове: Системата позволява на преподавателите да определят крайни срокове за тестове и задачи, което може да помогне да се гарантира, че студентите ще завършат курсовата си работа навреме.
3. Резултати: Системата записва и съхранява резултатите от тестове и задачи, които могат да се използват от преподавателите за проследяване на напредъка на учениците и идентифициране на области, в които може да е необходима допълнителна подкрепа.
4. Обратна връзка и оценяване: Системата включва система за обратна връзка и оценяване, където преподавателите могат да предоставят обратна връзка и оценки за тестове и задачи.
5. Персонализиране: Системата е много адаптивна, с функции, които могат да се добавят или премахват в зависимост от нуждите на учебното заведение.
6. Отчитане: Системата включва функция за отчитане, където преподавателите могат да генерират отчети за представянето на учениците, резултатите от тестовете и задачите и други подходящи данни.

Обхватът на проекта за училищна система е широк, с фокус върху предоставянето на цялостна платформа за образователните институции, за да управляват своите процеси на тестване, задания и уроци. Системата е проектирана да бъде мащабируема и гъвкава, с функции, които могат да бъдат персонализирани, за да отговарят на нуждите на различни институции и потребители. Освен това системата е достъпна от различни устройства, включително настолни компютри, лаптопи, таблети и смартфони, което улеснява достъпа на потребителите до системата от всяко място с интернет връзка.

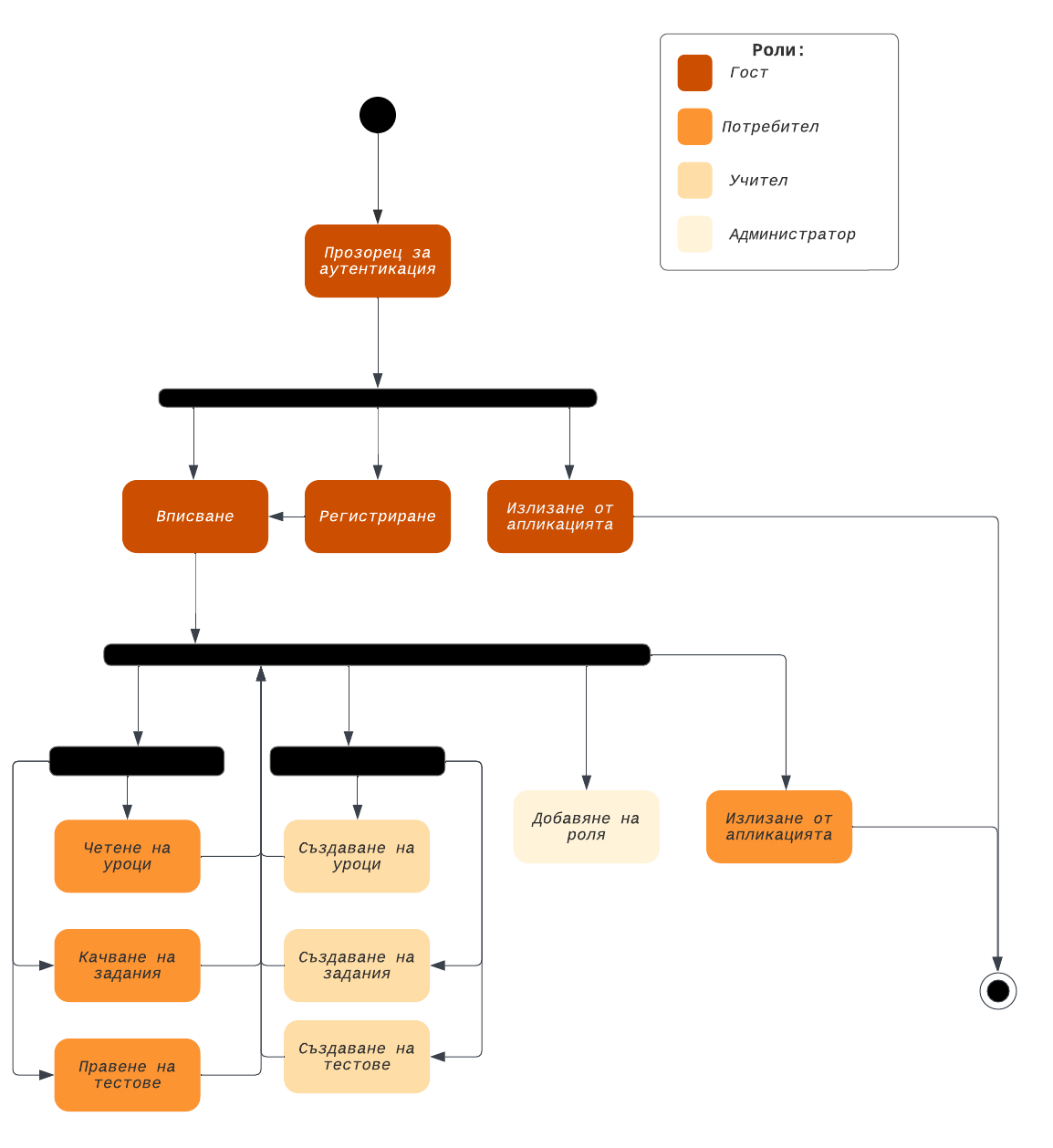
Едно от ключовите предимства на проекта е способността му да рационализира и автоматизира много от процесите, включени в тестването, подаването на задачи и достъпа до уроците. Като предоставя централизирана платформа за тези дейности, системата може да помогне за намаляване на административните разходи, като същевременно предоставя по-ангажиращо и интерактивно учебно изживяване за учениците.

Като цяло обхватът на проекта за училищна система е широкообхватен, включващ разнообразие от характеристики и функции, които са предназначени да отговорят на нуждите на образователните институции, преподавателите и учениците. Със своя гъвкав и мащабируем дизайн, системата може да бъде персонализирана, за да отговори на нуждите на различни институции и потребители, като същевременно предоставя удобна за потребителя и ангажираща платформа за управление на тестове, задачи и уроци.

# Анализ на решението

## Потребителски изисквания и работен процес

### Потребителски изисквания



**Фигура 3.1.1.1**

Един от ключовите аспекти на проектирането на ефективно софтуерно приложение е разбирането на нуждите и изискванията на неговите потребители. За да се гарантира, че проектът отговаря на нуждите на предвидените потребители, от съществено значение е да идентифицирате и документирате тези изисквания.

Като начало, първо трябва да идентифицираме различните типове потребители, които ще взаимодействат със системата. Проектът ще се използва от четири основни типа потребители: гости, потребители, учители и администратори.

Гостите са лица, които все още не са създали акаунт в системата. Те могат да преглеждат съдържанието на уебсайта, но нямат достъп до нито една от функционалностите на системата. Гостите могат да създадат акаунт или да влязат с помощта на акаунти в социалните медии.

Потребителите са лица, които са създали акаунт и имат достъп до функционалностите на системата. Потребителите могат да четат уроци, да правят тестове и да качват задачи. Те също могат да видят своите оценки и напредък в системата. Потребителите могат да бъдат студенти или участници в онлайн курсове.

Учителите са лица, които имат акаунт в системата и могат да създават и управляват курсове, уроци, тестове и задачи. Учителите могат да виждат напредъка на своите ученици и да оценяват работата им.

Администраторите са лица, които имат достъп до административните функции на системата, които включват управление на потребителски акаунти, настройка на роли и наблюдение на използването на системата. Администраторите могат също да управляват и модерират съдържанието на системата.

След като сме идентифицирали типовете потребители, можем да преминем към дефиниране на различните случаи на употреба, които системата трябва да поддържа, за да отговори на нуждите на своите потребители. Случаите на употреба помагат да се идентифицират различните потребителски действия, необходими за изпълнение на конкретни задачи в системата.

За гостите случаите на използване са относително прости. Те могат да видят съдържанието на уебсайта, да се регистрират за акаунт или да влязат с помощта на акаунти в социални медии. За потребителите случаите на използване са по-сложни. Потребителите могат да четат уроци, да правят тестове и да качват задачи. Те могат да видят напредъка и оценките си, както и каталога на курсовете и записаните курсове.

За учителите случаите на използване включват създаване и управление на курсове, уроци, тестове и задачи. Учителите също могат да преглеждат напредъка на своите ученици и да оценяват работата им. Администраторите могат да управляват потребителски акаунти, роли и разрешения. Те могат също така да преглеждат данни за използването на системата и съдържание.

След като сме идентифицирали различните случаи на употреба, необходими за всеки тип потребител, можем да преминем към дефиниране на функционалните изисквания, необходими за поддържане на тези случаи на употреба. Това ще помогне да се гарантира, че системата е проектирана да отговаря на нуждите на своите потребители и да им предоставя функционалностите, от които се нуждаят, за да изпълняват задачите си ефикасно и ефективно.

В обобщение, разбирането на изискванията на потребителите е от решаващо значение за разработването на успешно софтуерно приложение. Чрез идентифициране на различните типове потребители, дефиниране на различните случаи на употреба, които те ще трябва да поддържат, и дефиниране на функционалните изисквания за поддръжка на тези случаи на употреба, можем да гарантираме, че отговаря на нуждите на предвидените потребители.

### Работен процес

Работният процес на софтуера може да бъде описан по отношение на вход, обработка и изход. Входът е искането на потребителя за конкретно действие, като например създаване на тест или качване на задание. Този вход може да дойде от всеки от участниците в системата, включително гост, потребител, учител или администратор.

След като входът бъде получен, системата го обработва, като проверява идентификационните данни на потребителя и проверява дали има необходимите разрешения за извършване на исканото действие. Ако потребителят е оторизиран, системата изпълнява действието, като създава нов тест, добавя задание към базата данни или създава нов урок.

След като обработката приключи, системата генерира изход, който може да включва съобщение за потвърждение, списък с налични тестове или задачи или съдържанието на урок. Резултатът се представя на потребителя чрез потребителския интерфейс, който е проектиран да бъде интуитивен и удобен за потребителя.

За да осигурим по-добро разбиране на работния процес, ние създадохме диаграма на случаи на използване, която илюстрира различните действия, които могат да бъдат извършени от участниците в системата. Диаграмата показва различните случаи на употреба, достъпни за всеки участник, както и връзките между случаите на употреба.

Например, гостът може да влезе, да се регистрира и да преглежда курсове. След като потребителят влезе в системата, той може да чете уроци, да опитва тестове и да качва задачи. Учителят може да създава уроци, тестове и задачи, докато администраторът може да управлява потребителски роли и разрешения.

В допълнение към диаграмата на случаите на употреба, създадохме и диаграми на дейността, които показват последователността от стъпки, включени в конкретни случаи на употреба, като например създаване на нов тест или качване на задание. Тези диаграми предоставят по-подробен изглед на работния процес и могат да се използват за идентифициране на потенциални затруднения или области за подобрение.

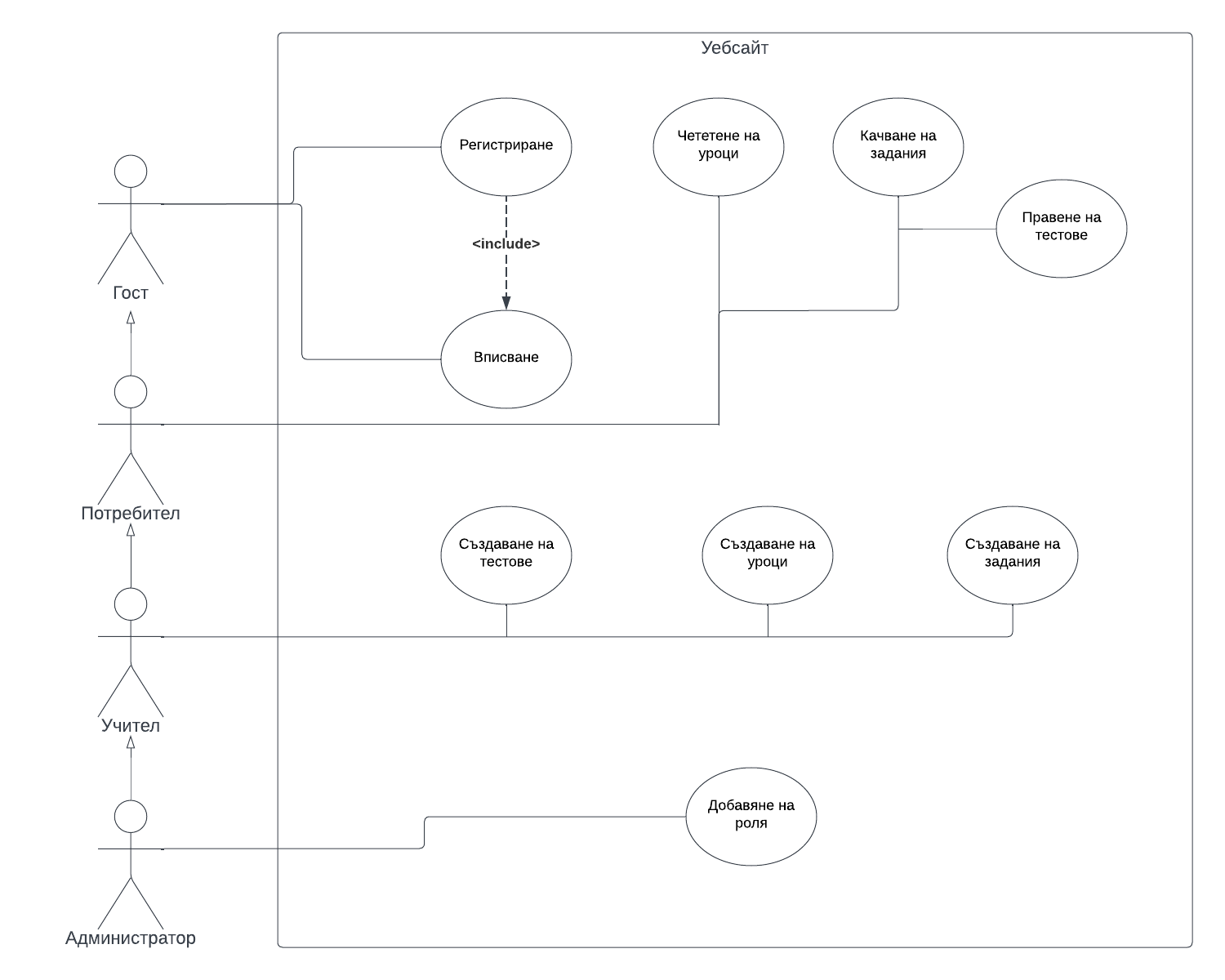
Като цяло софтуерът е проектиран да осигурява безпроблемен и ефективен работен процес за потребителите, независимо от тяхната роля в системата. Следвайки ясен модел на въвеждане, обработка и изход, системата може да осигури последователно и надеждно изживяване за потребителите, като същевременно минимизира потенциала за грешки или объркване.

## Примерен потребителски интерфейс

Допълнете резултатите от статистически анализа на проблема, описани в секция 3.1, с фигури на примерен графичен интерфейс /създадени или в самата среда заедно с потоците от събития, или извън нея/.

## Диаграми на анализа

### Диаграма на случаите (Uses Cases Diagram)



**Фигура 3.3.1.1**

Диаграмата включва четири актьора, които са „Гост“, „Потребител“, „Учител“ и „Администратор“, всеки със своите съответни случаи на употреба.

Гост-актьорът има достъп само до три случая на употреба, които са „Вход“, „Регистрация“ и „Преглед на курсове“. Случаят на използване „Вход“ позволява на госта да влезе в системата със своето потребителско име и парола, докато случаят на използване „Регистрация“ им позволява да създадат нов акаунт. Случаят на използване на „Преглед на курсове“ позволява на госта да преглежда наличните курсове в системата.

Актьорът потребител има достъп до седем случая на използване, които са „Четене на уроци“, „Опит за тестове“, „Качване на задания“, „Преглед на резултатите“, „Промяна на парола“, „Преглед на профил“ и „Изход“. Случаят на използване „Прочетете уроци“ позволява на потребителя да има достъп и да чете наличните материали за курса, докато случаят на използване на „Опит за тестове“ им позволява да вземат тестове, зададени от техния учител. Случаят на използване на „Качване на задачи“ позволява на потребителя да изпрати своите задачи на своя учител за оценка. Случаят на използване на „Преглед на резултатите“ позволява на потребителя да преглежда своите резултати от тестове и оценки от задачите. Случаят на използване на „Промяна на паролата“ позволява на потребителя да актуализира своята парола. Случаят на използване „Преглед на профила“ позволява на потребителя да види личната си информация, а случаят на използване „Изход“ му позволява да излезе от системата.

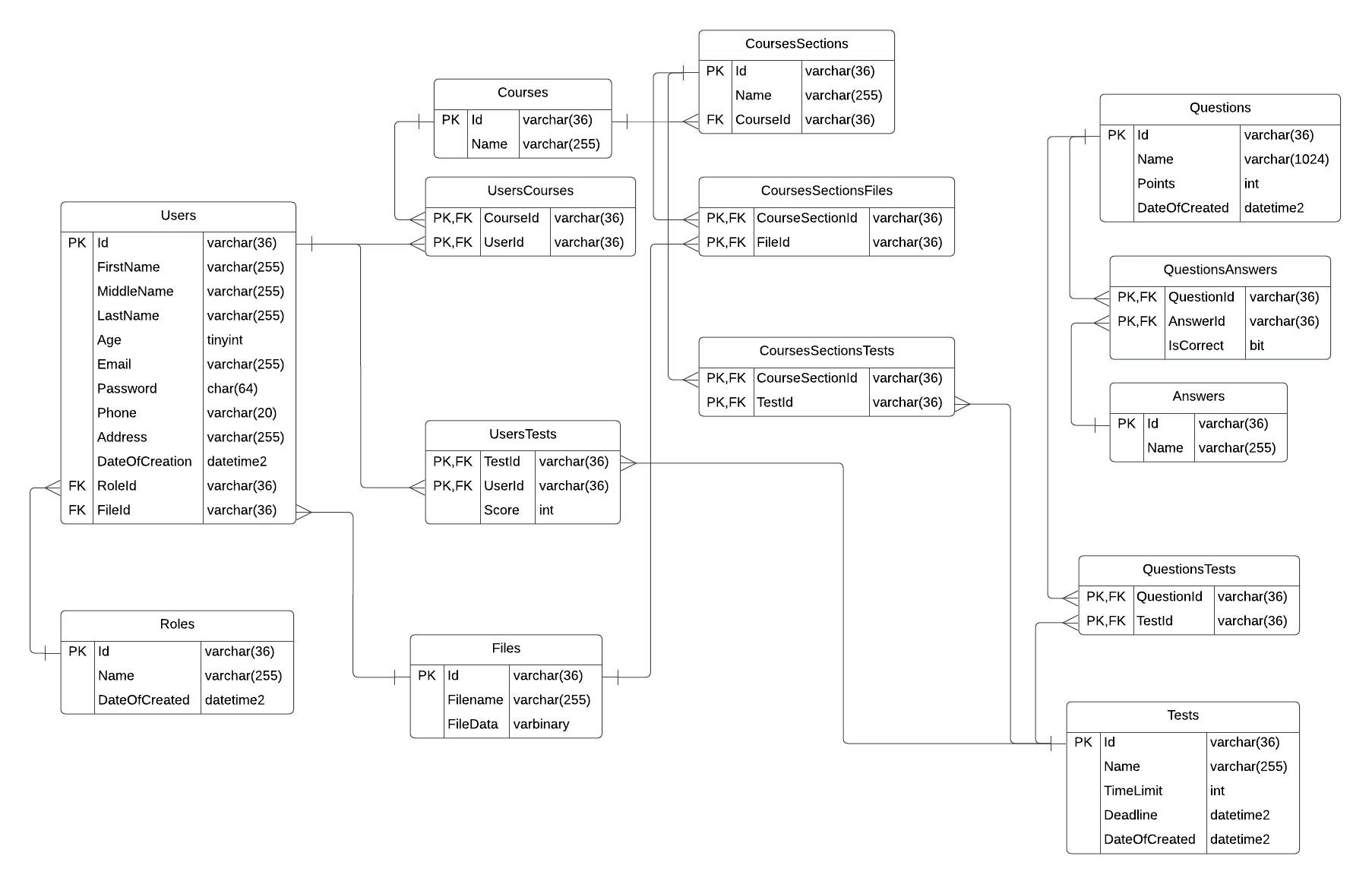
Актьорът Учителят има достъп до шест случая на използване, които са „Създаване на уроци“, „Създаване на тестове“, „Създаване на задания“, „Преглед на резултатите“, „Промяна на парола“ и „Изход“. Случаят на използване на „Създаване на уроци“ позволява на учителя да създава и публикува нови учебни материали за своите ученици. Случаят на използване на „Създаване на тестове“ позволява на учителя да създава и възлага тестове, които учениците да вземат. Случаят на използване на „Създаване на задачи“ позволява на учителя да създава и възлага нови задачи на своите ученици. Случаят на използване на „Преглед на резултатите“ позволява на учителя да преглежда резултатите от тестовете на своите ученици и оценките от задачите. Случаят на използване на „Промяна на паролата“ позволява на учителя да актуализира своята парола. Случаят на използване „Излизане“ позволява на учителя да излезе от системата.

Актьорът на администратора има достъп само до един случай на употреба, който е „Управление на роли“. Случаят на използване на „Управление на роли“ позволява на администратора да присвоява роли на различните участници в системата, като например предоставяне на учителски или администраторски привилегии на конкретни потребители.

Диаграмата на случаите на използване също така включва редица взаимоотношения между участниците и техните съответни случаи на употреба. Например, случаят на използване „Влизане“ е свързан както с актьорите „Гост“, така и с „Потребител“, което показва, че гостите могат да станат потребители, като влязат. По подобен начин случаите на използване „Извършване на тестове“ и „Създаване на задания“ са свързани към актьорите „Учител“ и „Потребител“, което показва, че учителите и потребителите могат да създават и възлагат тестове и задачи.

Като цяло вашата диаграма на случаите на използване предоставя ясен преглед на различните участници и техните взаимодействия със софтуерното приложение School System. Той подчертава ключовите случаи на употреба за всеки участник и техните взаимоотношения един с друг. Това ще бъде ценно за екипа за разработка, докато работят за внедряване на различните функции на системата, и за заинтересованите страни, когато се стремят да разберат функционалността на софтуерното приложение.

### ER диаграма



**Фигура 3.3.2.1**

ER диаграмата, предоставена за проекта, е нормализирана до третата нормална форма (3NF). Процесът на нормализиране включва премахването на излишните данни и организирането на данните в логически и смислени групи.

В първата нормална форма (1NF), ER диаграмата гарантира, че всеки атрибут има една стойност. Всеки обект има уникален идентификатор и атрибутите във всеки обект са атомарни. Всички обекти в диаграмата отговарят на първата нормална форма. Например обектът Users има уникален идентификатор „Id“ и всички негови атрибути са атомарни. Всеки атрибут представлява една стойност и не се дублира.

Във втората нормална форма (2NF), ER диаграмата гарантира, че всички неключови атрибути зависят от целия първичен ключ. Втората нормална форма се постига чрез разделяне на обекти със съставни първични ключове на отделни обекти с атомарни ключове. Диаграмата също така гарантира, че всеки неключов атрибут зависи от целия първичен ключ. Например обектът Users има съставен първичен ключ, включващ Id, RoleId и FileId. Всеки неключов атрибут, като FirstName, MiddleName и LastName, зависи от целия първичен ключ.

В третата нормална форма (3NF), ER диаграмата гарантира, че всички неключови атрибути са независими един от друг. Третата нормална форма се постига чрез елиминиране на преходни зависимости между неключови атрибути. Диаграмата гарантира, че всички неключови атрибути зависят само от първичния ключ, а не от други неключови атрибути.

Например, обектът Courses има уникален идентификатор „Id“ и всички негови атрибути са атомарни. Обектът CourseSections има външен ключ „CourseId“, който препраща към първичния ключ „Id“ на обекта Course. Атрибутът Name на обекта CourseSections зависи само от първичния ключ „Id“, а не от който и да е друг атрибут на обекта CourseSections. Обектът CoursesSections също има уникален идентификатор „Id“ и всички негови атрибути са атомарни.

В обобщение, ER диаграмата, предоставена за проекта, е нормализирана до третата нормална форма, което гарантира премахването на всички транзитивни зависимости между неключови атрибути. Процесът на нормализиране гарантира, че данните в диаграмата са добре организирани, намалява излишъка и свежда до минимум риска от несъответствия в данните.

## Модел на съдържанието / данните

Тук опишете модела на данните/съдържанието - текстово, графично и евентуално аудио/видео съдържание), което ще представите в проекта си. Опишете размера и типа на данните/файловете и начина на кодиране за всеки от ресурсите.

Ако ползвате в проекта си текстово и/или мултимедийно съдържание от различни типове, представете неговата структура, напр. посредством таксономия, типология, онтология или други схеми за представяне на структурата от категории, под-категории, типове и т.н., както и техните взаимовръзки с други категории или типове, напр. географски региони и дялове, области/сегменти от промишлеността, и др.

След структурата опишете и възможните характеристики, атрибути и честота на срещане на всеки един ресурс в съдържанието (категория, тип, екземпляр, връзка/релация и т.н.).

# Дизайн

Тази секция представя дизайна на решението на проблема за проекта ви. Опишете каква софтуерна платформа сте избрали за вашето решение /напр. .NET, java/. Представете схема на софтуерната архитектура на решението /по модули и/или слоеве/ с диаграма на разгръщането, както и диаграми на класовете на дизайна /с ограничения, описани на OCL/, диаграми на времето /за задаване на времена за синхронизация и комуникация в решението/ и компонентни диаграми. Илюстрирайте решението с извадки от генериран сорс код.

## Реализация на архитектурата на приложението

Проектът е реализиран с помощта на архитектурния модел Model-View-Controller (MVC). Този модел разделя приложението на три отделни слоя: модели, изгледи и контролери.

### Модели

Моделният слой отговаря за управлението на данните и бизнес логиката на приложението. В проекта слоят Model (Модел) е имплементиран с помощта на Entity Framework, която е рамка за ORM (обектно релационно картографиране) за .NET приложения. Entity Framework опростява достъпа до данни, като позволява на разработчиците да взаимодействат с базата данни, използвайки техники за обектно-ориентирано програмиране, вместо SQL заявки.

Моделният слой в проекта се състои от обекти и хранилища. Субектите представляват моделите на домейни, като тестове, въпроси, задания и уроци, докато хранилищата са отговорни за запазването и извличането на данните от базата данни. Репозиториите комуникират с базата данни с помощта на Entity Framework, която обработва всички операции на базата данни на ниско ниво, като създаване на таблици, заявки за данни и актуализиране на записи.

### Изгледи

Презентационния слой е отговорен за представянето на данните на потребителя по ясен и удобен за потребителя начин. В проекта презентационния слой се реализира с помощта на HTML, CSS и JavaScript. Потребителският интерфейс е проектиран с помощта на CSS рамката Bootstrap, която осигурява адаптивен дизайн и предварително изградени UI компоненти, които могат лесно да бъдат персонализирани, за да отговарят на нуждите на приложението. Презентационният слой също използва синтаксиса за маркиране на Razor, който се използва за създаване на динамични уеб страници с код от страна на сървъра.

Потребителският интерфейс в проекта е проектиран да бъде прост и интуитивен, с ясни навигационни менюта и лесни за използване формуляри за създаване и управление на тестове, задачи и уроци. Потребителският интерфейс също така включва визуални знаци, като икони и цветно кодирани елементи, за да улесни потребителите да разбират и взаимодействат със системата.

### Контролери

Слоят на контролера е отговорен за обработката на потребителски заявки, обработката на потребителски вход и комуникацията със слоя модел за извличане и актуализиране на данни. В проекта слоят контролер е имплементиран с помощта на рамката ASP.NET MVC, която предоставя рамка за изграждане на уеб приложения, използвайки модела на архитектурата Model-View-Controller.

Слоят на контролера в проекта е отговорен за обработката на потребителски заявки, обработката на потребителски вход и комуникацията със слоя модел за извличане и актуализиране на данни. Слоят на контролера също така управлява маршрутизирането, което съпоставя URL адресите с действията на контролера, и валидирането на входа, което гарантира, че въведеното от потребителя е валидно, преди да бъде обработено от слоя модел.

Като цяло, внедряването на архитектурата на приложението в проекта осигурява гъвкава, мащабируема и ефективна платформа за управление на образователни тестове, задачи, уроци и други данни. Чрез използването на модела на архитектурата Model-View-Controller, проектът е проектиран да бъде модулен и лесен за поддръжка. Чрез използването на Entity Framework, Bootstrap и ASP.NET MVC разработката на приложението е рационализирана, ефективна и мащабируема.

## Описание на слоевете, предназначението им, библиотеки и методи включени в съответния слой.

Проектът е проектиран с помощта на модела на архитектурата MVC (Model-View-Controller). Този модел разделя приложението на три слоя:

1. Слой модел
2. Презентационен слой
3. Контролен слой

### Слой модел

Моделният слой е отговорен за управлението на данните и бизнес логиката на приложението. Състои се от обекти и хранилища. Обектите представляват моделите на домейни, а хранилищата са отговорни за запазването и извличането на данните от базата данни. Целта на моделния слой е да осигури ясно разделение между логиката на приложението и базата данни.

Библиотеките, използвани в моделния слой, включват:

* Entity Framework: Entity Framework е популярна рамка за ORM (обектно релационно картографиране) за .NET приложения. Той опростява достъпа до данни и осигурява по-обектно-ориентиран начин за взаимодействие с базата данни.
* LINQ: LINQ е интегрирана в езика заявка, която позволява на разработчиците да пишат сложни заявки към базата данни, използвайки прост синтаксис.

Методите за включване в моделния слой включват:

* Достъп до данни: Моделният слой е отговорен за обработката на всички задачи за достъп до данни. Това включва извличане на данни от базата данни, вмъкване на нови данни, актуализиране на съществуващи данни и изтриване на данни.
* Бизнес логика: Моделният слой е отговорен и за внедряването на бизнес логиката на приложението. Това включва налагане на правила за валидиране на данни, управление на взаимоотношения между обекти и прилагане на бизнес правила.

### Презентационен слой

Презентационния слой е отговорен за представянето на данните на потребителя по ясен и удобен за потребителя начин. Състои се от HTML страници и UI елементи, с които потребителят взаимодейства. Целта на презентационен слой е да предостави потребителски интерфейс за приложението, който е лесен за използване и разбиране.

Библиотеките, използвани в слоя за преглед, включват:

* Bootstrap: Bootstrap е популярна CSS рамка, която предоставя отзивчив дизайн и предварително изградени компоненти на потребителския интерфейс, които могат лесно да бъдат персонализирани, за да отговарят на нуждите на приложението.
* jQuery: jQuery е популярна JavaScript библиотека, която опростява манипулирането на DOM, обработката на събития и Ajax заявките.

Методите за включване в слоя за преглед включват:

* Дизайн на потребителския интерфейс: Презентационен слой е отговорен за проектирането на потребителския интерфейс на приложението. Това включва създаване на HTML страници, проектиране на формуляри и създаване на UI елементи, които са лесни за използване и разбиране.
* Потребителско въвеждане: Слоят за преглед също е отговорен за обработката на въведеното от потребителя. Това включва валидиране на въведеното от потребителя, обработка на подавания на формуляри и показване на съобщения за грешка на потребителя, когато е необходимо.

### Контролен слой

Слоят на контролера е отговорен за обработката на потребителски заявки, обработката на потребителски вход и комуникацията с моделния слой за извличане и актуализиране на данни. Целта на контролерния слой е да действа като посредник между потребителя и моделния слой.

Библиотеките, използвани в контролерния слой, включват:

* ASP.NET MVC: ASP.NET MVC е рамка за изграждане на уеб приложения, използвайки модела на архитектурата Model-View-Controller.
* Razor: Razor е синтаксис за маркиране, използван за създаване на динамични уеб страници с код от страна на сървъра.

Методите за включване в контролерния слой включват:

* Маршрутизиране: Слоят на контролера е отговорен за обработката на потребителските заявки и насочването им към подходящото действие на контролера.
* Обработка на въвеждане от потребителя: Слоят на контролера е отговорен за обработката на въведеното от потребителя и валидирането му, преди да го изпрати до слоя модел.
* Извличане и актуализиране на данни: Слоят на контролера е отговорен за комуникацията със слоя на модела за извличане и актуализиране на данни, ако е необходимо.

Като цяло, слоевете в проекта осигуряват ясно разделение на проблемите и улесняват управлението на данните и бизнес логиката на приложението. Чрез използването на библиотеки като Entity Framework, Bootstrap и jQuery разработката на приложението е рационализирана, ефективна и мащабируема.

## Организация и код на заявките към база от данни

Проектът е изграден с помощта на рамката ASP.NET, която предоставя различни инструменти и технологии за изграждане на уеб приложения. Един от ключовите компоненти на проекта е базата данни, която се използва за съхраняване на информация за тестове, въпроси, задачи, резултати и други подходящи данни. За достъп до базата данни проектът използва инструментариум за достъп до база данни, който предоставя методи за извличане, добавяне и изтриване на обекти в базата данни.

От гледна точка на програмния код, инструментариумът за достъп до база данни за проекта е изграден с помощта на Entity Framework, популярен инструмент за ORM (обектно релационно картографиране) инструмент за .NET приложения. Entity Framework позволява на разработчиците да работят с бази данни, използвайки техники за обектно-ориентирано програмиране, вместо директно да пишат SQL заявки на ниско ниво.

За да използват Entity Framework в проекта , разработчиците създават клас контекст на данни, който наследява от класа DbContext, предоставен от рамката. Този клас на контекст на данни представлява връзката към базата данни и включва свойства, които се преобразуват в таблици на база данни.

Например, за да извлече списък с тестове от базата данни, разработчикът може да създаде метод като следния:

public List<Test> GetTests()

{

using (var context = new SchoolSystemContext())

{

return context.Tests.ToList();

}

}

В този метод класът SchoolSystemContext представлява връзката към базата данни, а свойството тестове представлява таблицата с тестове в базата данни. Методът използва синтаксиса LINQ to Entities, за да направи заявка в базата данни и да върне списък с тестови обекти.

По същия начин, за да добави нов тест към базата данни, разработчикът може да създаде метод като следния:

public void AddTest(Test test)

{

using (var context = new SchoolSystemContext())

{

context.Tests.Add(test);

context.SaveChanges();

}

}

В този метод методът Add() се използва за добавяне на новия тестов обект към таблицата тестове в базата данни, а методът SaveChanges() се използва за записване на промените в базата данни.

И накрая, за да изтрие тест от базата данни, разработчикът може да създаде метод като следния:

public void DeleteTest(Test test)

{

using (var context = new SchoolSystemContext())

{

context.Tests.Remove(test);

context.SaveChanges();

}

}

В този метод методът Remove() се използва за изтриване на посочения тестов обект от таблицата тестове в базата данни, а методът SaveChanges() се използва за записване на промените в базата данни.

Тези методи предоставят основен пример за инструментариум за достъп до база данни за проекта. На практика може да има много повече включени методи и класове в зависимост от сложността на приложението и изискванията на образователната институция.

Като цяло използването на Entity Framework в проекта предоставя гъвкав и мощен инструментариум за работа с базата данни. Чрез използване на техники за обектно-ориентирано програмиране, вместо директно писане на SQL заявки на ниско ниво, разработчиците могат да работят по-ефективно и ефективно, като същевременно се възползват от функциите и функционалността, предоставени от Entity Framework. Освен това методите за извличане, добавяне и изтриване на обекти в базата данни са лесни за използване и разбиране, което помага да се гарантира, че базата данни се управлява ефективно и ефикасно.

## Наличие на потребителски интерфейс (конзолен, графичен, уеб)

Описание на основните функционалности на интерфейса на приложението.

Забележка: Няма формално изискване на определен брой диаграми от даден вид, за даден брой проектанти.

# Ефективност и бързодействие на решението

Ефективността и скоростта на вземане на решения са критични фактори за образователните институции, които трябва да управляват големи обеми тестове, задачи и уроци. Проектът „Училищна система“ е проектиран с оглед на тези фактори, използвайки различни характеристики и функции, които могат да помогнат за подобряване на ефективността и скоростта на вземане на решения от преподаватели и администратори.

За да се анализира ефективността и скоростта на вземане на решения за проекта за училищна система, е полезно да се разгледат известните решения и съответните литературни източници за управление на тестове, задачи и уроци в образователни институции. Ето няколко примера за известни решения и литературни източници:

1. Хартиени системи: Традиционните хартиени системи се използват от много години за управление на тестове, задачи и уроци в образователните институции. Въпреки че тези системи са познати и сравнително евтини, те могат да бъдат бавни и неефективни, изискващи ръчно оценяване, водене на записи и отчитане.
2. Системи за управление на обучението (LMS): Системите за управление на обучението, като Blackboard и Canvas, обикновено се използват в образователни институции за управление на онлайн курсове, включително тестове, задачи и уроци. Тези системи могат да бъдат по-ефективни от системите на хартиен носител, тъй като позволяват автоматизирано оценяване, водене на записи и отчитане. Въпреки това, LMS могат да бъдат скъпи и сложни за настройка и поддръжка и може да не предоставят гъвкавостта и опциите за персонализиране, които някои образователни институции изискват.
3. Онлайн инструменти за оценяване: Налични са различни онлайн инструменти за оценяване, като Kahoot, Quizlet и Google Forms, които могат да се използват за създаване и управление на тестове и задачи. Тези инструменти могат да бъдат относително евтини и лесни за използване, с функции като автоматизирано оценяване и отчитане. Те обаче може да не предоставят гъвкавостта и опциите за персонализиране, които някои образователни институции изискват, и може да не са подходящи за управление на големи обеми тестове и задачи.
4. Изградени по поръчка системи: Някои образователни институции избират да разработят изградени по поръчка системи за управление на тестове, задачи и уроци. Тези системи могат да бъдат силно персонализирани и гъвкави, с функции, които отговарят на специфичните нужди на институцията. Въпреки това, изградените по поръчка системи могат да бъдат скъпи и отнемащи време за разработване и поддръжка и може да не са мащабируеми за по-големи институции или по-големи обеми от данни.

Като се имат предвид тези решения и литературни източници, проектът за училищна система е предназначен да осигури баланс между ефективност, скорост на вземане на решения и гъвкавост. Системата е изградена с помощта на рамката ASP.NET и използва база данни за съхраняване на информация за тестове, въпроси и резултати. Този подход, управляван от база данни, позволява автоматизирано оценяване, водене на записи и отчитане, което може да помогне за подобряване на ефективността и скоростта на вземане на решения от преподаватели и администратори.

Освен това проектът включва разнообразие от характеристики и функции, които са предназначени да рационализират и автоматизират много от процесите, включени в управлението на тестове, задачи и уроци. Например, преподавателите могат да създават и управляват тестове и задачи онлайн, с възможността да персонализират инструкции, критерии за оценяване и крайни срокове. Учениците имат достъп до тестовете и задачите онлайн, да ги попълнят и да ги изпратят за оценяване и обратна връзка, като резултатите се записват в базата данни за по-късен преглед.

За по-нататъшно подобряване на ефективността и скоростта на вземане на решения проектът включва различни инструменти за отчитане и анализ. Преподавателите могат да генерират отчети за представянето на учениците, резултатите от тестовете и заданията и други подходящи данни, които могат да се използват за информиране на решенията за преподаване и учебна програма. Освен това системата включва система за обратна връзка и оценяване, където преподавателите могат да предоставят обратна връзка и оценки за тестове и задачи, което спомага за улесняване на по-бързото вземане на решения и по-ефективен учебен процес.

Като цяло проектът е предназначен да осигури ефективна и ефективна платформа за управление на тестове, задачи и уроци в образователните институции. Чрез използване на подход, управляван от база данни, рационализиране и автоматизиране на процесите и предоставяне на инструменти за отчитане и анализ, системата може да помогне за подобряване на ефективността и скоростта на вземане на решения за преподаватели и администратори, като същевременно предоставя по-ангажиращо и интерактивно учебно изживяване за студенти.

# Тестване

Тук се *включват тестовите случаи* и какви видове тестване предвиждате в реалното изпълнение на проекта, напр. с колко и какви документи, в какви браузъри, с какви приставки, и т.н.

# Заключение и възможно бъдещо развитие

## Заключение

Проектът е цялостно софтуерно приложение, предназначено да улесни управлението на тестове, задачи и уроци за образователни институции. Проектът използва различни технологии и езици за програмиране, включително рамката ASP.NET, Entity Framework и SQL Server, за да осигури гъвкава и мощна платформа за образователните институции, за да управляват техните процеси на тестване, задания и уроци. Проектът е проектиран с оглед на ефективността и бързината на вземане на решения, като използва функции като автоматизирано оценяване, отчитане и инструменти за анализ, за да помогне на преподавателите и администраторите да управляват по-ефективно своите работни процеси. Като цяло проектът е изключително ефективно решение за управление на образователни тестове, задачи и уроци и има потенциала да осигури значителни ползи за образователните институции и техните ученици.

Предимствата на технологиите, езиците за програмиране и методите, използвани в проекта за училищна система, включват:

1. Гъвкавост: ASP.NET Framework и Entity Framework предоставят гъвкава и мащабируема платформа за изграждане на уеб приложения, което позволява персонализиране и адаптивност, за да отговори на нуждите на различни образователни институции.
2. Ефективност: Използването на инструменти за автоматизирано оценяване, отчитане и анализ може да помогне за подобряване на ефективността и скоростта на вземане на решения от преподаватели и администратори.
3. Достъпност: Възможността за достъп до проекта от различни устройства, включително настолни компютри, лаптопи, таблети и смартфони, улеснява достъпа на потребителите до системата от всяко място с интернет връзка.

Използваемостта на решението на практика е висока, предоставяйки гъвкава, ефективна и мощна платформа за управление на образователни тестове, задачи и уроци. За да се осигури постоянна ефективност на решението, е важно проектът да се прецизира и подобрява с течение на времето.

Като цяло проектът предоставя изключително ефективно решение за управление на образователни тестове, задачи и уроци. Чрез използване на подход, управляван от база данни, рационализиране и автоматизиране на процесите и предоставяне на инструменти за отчитане и анализ, системата може да помогне за подобряване на ефективността и скоростта на вземане на решения за преподаватели и администратори, като същевременно предоставя по-ангажиращо и интерактивно учебно изживяване за студенти. Тъй като проектът непрекъснато се усъвършенства и подобрява, той има потенциала да се превърне в основен инструмент за образователните институции по целия свят.

## Възможно бъдещо развитие

Проектът „Училищна система“ има потенциал за няколко бъдещи развития. Ето някои потенциални области за бъдещо развитие:

1. Интеграция със системи за управление на обучението: Може да се разработи интеграция със съществуващи системи за управление на обучението, използвани от образователните институции. Това ще улесни преподавателите да управляват плановете си за уроци и ще гарантира, че техните ученици имат достъп до необходимите материали.
2. Интегриране с информационни системи за студенти: Интегрирането с информационни системи за студенти би помогнало за автоматизирането на записването на студентите и проследяването на оценките. Това ще помогне на преподавателите да спестят време и да намалят ръчните грешки.
3. Мобилна оптимизация: Може да се разработи допълнително оптимизиране на проекта за училищна система за мобилна употреба, за да стане по-удобен за ученици и учители, които имат достъп до системата от мобилни устройства.
4. Усъвършенстван анализ и отчитане: Усъвършенствани инструменти за анализ и отчитане могат да бъдат разработени, за да осигурят по-подробна информация за представянето на учениците, позволявайки на преподавателите да коригират своите стратегии за преподаване в реално време.
5. Персонализирано обучение: Проектът за училищна система може да бъде разширен, за да позволи персонализирано обучение, като позволи на учениците да определят свои собствени учебни цели и съответно да коригират темпото си на обучение. Това ще помогне да се гарантира, че всеки ученик учи с удобно за него темпо.
6. Геймификация: Елементите на геймификация могат да бъдат включени в проекта за училищна система, за да стане по-ангажиращ и забавен за учениците. Това може да включва награди и значки за завършване на уроци или постигане на определени оценки.

Като цяло, тези бъдещи разработки имат потенциала да подобрят ефективността на проекта за училищна система, като го направят по-лесен за използване, ефикасен и ефективен. С напредването на технологиите е вероятно тези разработки да станат по-често срещани в образователните софтуерни приложения.

# Използвани литературни източници и Уеб сайтове

Използвайте вградената функционалност на Word: References > Citations & Bibliography

1. Уеб сайт на ….., адрес ….
2. Уеб сайт на ….., адрес ….
3. Уеб сайт на ….., адрес ….
4. Уеб сайт на ….., адрес ….
5. Литературен източник 2
6. Литературен източник 3
7. Литературен източник 4
8. Литературен източник 5

# Приложения

При необходимост можете да добавите и допълнителни секции под формата на апендикси. Таблица с диаграми, таблици и графики

Забележка:

1. Документацията на проекта се предава само в електронен вид в MS Word, чрез качването на архив с документа и останалите файлове по проекта, в задание за предаване на проект, в канала на екипа в Teams.
2. Кода на проекта, базата данни и документацията трябва да са налични в репозитори в GitHub, което е копие на заданието генерирано в организацията.

# Критерии и показатели за оценяване

|  |  |  |
| --- | --- | --- |
| Критерии и показатели за оценяване | Максимален брой точки за показателите | Максимален брой точки за критерия |
| 1. Съответствие с изискванията за съдържание и структура на дипломния проект |  | 20 |
| 1. 1. логическа последователност и структура на изложението, балансиране на отделните части | 4 |  |
| 1.2. задълбоченост и пълнота при формулиране на обекта, предмета, целта и задачите в разработването на темата | 7 |  |
| 1.3. използване на подходящи изследователски методи | 4 |  |
| 1.4. стил и оформяне на дипломната работа (терминология, стил на писане, текстообработка и оформяне на фигури и таблици) | 5 |  |
| 2. Съответствие между поставените цели на дипломния проект и получените резултати |  | 20 |
| 2.1. изводите следват пряко от изложението, формулирани са ясно, решават поставените в началото на изследването цели и задачи и водят до убедителна защита на поставената теза | 10 |  |
| 2.2. оригиналност, значимост и актуалност на темата | 6 |  |
| 2.3. задълбоченост и обоснованост на предложенията и насоките | 4 |  |
| 3. Представяне на дипломния проект |  | 20 |
| 3.1. представянето на разработката по темата e ясно и точно | 5 |  |
| 3.2. онагледяване на експозето с:  а) презентация;  б) графични материали;  в) практически резултати;  г) компютърна мултимедийна симулация и анимация | 10 |  |
| 3.3. умения за презентиране | 5 |  |
| 4. Отговори на зададените въпроси от рецензента и/или членовете на комисията за защита на дипломен проект |  | 30 |
| 4.1. разбира същността на зададените въпроси и отговаря пълно, точно и убедително | 10 |  |
| 4.2. логически построени и точни отговори на зададените въпроси | 10 |  |
| 4.3. съдържателни и обосновани отговори на въпросите | 10 |  |
| 5. Използване на професионалната терминология, добър и ясен стил, обща езикова грамотност |  | 10 |
| 5.1. Правилно използване на професионалната терминология | 5 |  |
| 5.2. Ясен изказ и обща езикова грамотност | 5 |  |
| Общ брой точки: | Максимален бр. точки 100 | Максимален бр. точки 100 |