



ДИПЛОМЕН ПРОЕКТ

Тема: РАЗРАБОТКА НА УЕБ ПРИЛОЖЕНИЕ

.....

Професия: код 481030 „Приложен програмист“
Специалност: код 4810301 „Приложно програмиране“

Изготвил:
Трите имена на ученика
Ученик от XII^Б клас

Ръководител - консултант:
Имена на консултант

Перник, 2023 г.

СЪДЪРЖАНИЕ

УВОД	3
1. ПЪРВА ГЛАВА – ПРОУЧВАНЕ	5
1.1. Предпоставките за създаване на продукта	5
1.2. Преглед на технологии и езици за програмиране	5
1.3. Избор на технологии и езици за програмиране	5
2. ВТОРА ГЛАВА – ПРОЕКТИРАНЕ	7
2.1. Описание на изискванията	7
2.1.1. Общо описание на изискванията	7
2.1.2. Функционални изисквания	8
2.1.3. Нефункционални изисквания	8
2.2. Анализ на функционалните изисквания	9
2.2.1. Изисквания към правата на потребителите	9
2.2.2. Изисквания към функционалния дизайн	10
2.2.3. Потребителски истории (user stories)	11
2.2.4. Случаи на употреба (use cases)	12
2.2.5. UML диаграми за случаи на употреба	14
2.3. Анализ на нефункционалните изисквания	15
2.4. Проектиране	16
2.4.1. Представяне	16
2.4.1. Класове и взаимоотношения	17
2.4.2. Проектиране на база данни	18
2.5. Прототипи на потребителския интерфейс	19
3. ТРЕТА ГЛАВА – РЕАЛИЗАЦИЯ	20
3.1. Създаване на проект	20
3.2. Структура на базата от данни	20
3.3. Изграждане на уеб приложението	20
3.3.1. Архитектура на приложението	20
3.3.2. Контролери	20
3.3.3. Изгледи	20
3.3.4. Сървиси	20
3.3.5. Други технологии и функционалности	20
ЗАКЛЮЧЕНИЕ	21
ИЗПОЛЗВАНА ЛИТЕРАТУРА	22
ПРИЛОЖЕНИЕ	23

Съвет: генерирайте автоматично съдържанието!

УВОД

Кратко въведение в областта; обект и предмет на дипломния проект; описание на основните цели и задачи; актуалност и значимост на темата. /1-2 стр./

Кратко въведение в темата

Обект на дипломния проект - уеб приложение.

Предмет на дипломния проект - проектиране и разработка на уеб приложение за, базирано на . ASP.Net Core MVC на езика C#, което използва база от данни на MS SQL Server.

Цел на дипломния проект - целта на дипломния проект е да се проектира и разработи уеб приложение за, в което да може да се публикува информация за, а също така и да има функционалност за покупка на стоки и/или услуги, предлагани в сайта.

Целевата аудитория на уеб приложението са:

Приложението трябва да поддържа надеждна система за автентикация и авторизация на своите потребители, които трябва да бъдат разпределени минимум в три роли: нерегистрирани потребители (гости), регистрирани потребители (клиенти) и администратор. Допълнете други особености на приложението.

Уеб приложението трябва да се базира на съвременни технологии, стандарти и модели, което гарантира лесна бъдеща поддръжка, ефективност и възможност за надграждане.

Задачи за изпълнение:

За постигане на целите на дипломния проект ще бъдат изпълнени следните задачи:

- Проучване и анализ на възможностите на други подобни приложения
- Преглед на известните технологии, използвани в уеб разработката
- Избор на подходящи технологии
- Анализ на изискванията
- Проектиране и
- Реализация на приложението
 - Създаване на MVC проект
 - Създаване на базата от данни

- Изграждане на навигационна система на сайта
- Създаване на функционалностите на сайта: добавяне, изтриване, променяне и резервиране на продукт; преглед на собствените направени резервации; и т.н. .

Актуалност на проблема - **защо изграждането на уеб приложение за е актуална и значима тема.**

Софтуер за изграждане на уеб приложението - за изграждането на уеб приложението ще бъдат използвани: *Visual Studio Community 2019* и *MS SQL Server*. Уеб приложението ще бъде базирано на *ASP.NET Core* технологията на Microsoft, *MVC* моделът и езикът за програмиране *C#*. Ще бъдат използвани и други помощни софтуерни продукти, като *SQL Server Management Studio (SSMS)*, както и други програмни езици технологии, като *CSS*, *Razor*, *Bootstrap* и др.

Описват се тези технологии, които наистина сте използвали в проекта!

За разработката на софтуерния продукт ще преминем през три етапа:

- Проучване
- Проектиране
- Реализация

Трите етапи ще бъдат разгледани в следващите глави.

1.ПЪРВА ГЛАВА – ПРОУЧВАНЕ

Прави се преглед на съществуващи подобни сайтове или приложения и техните възможности. Разглеждат се известните технологии и развойни среди, осигуряващи създаването на приложението. Аргументирано се избират подходящи технологии /изводи за подходящите технологии/. /5-7 стр./

1.1. Предпоставките за създаване на продукта

Има много причини за създаване на софтуерен продукт. Неговата реализация е свързана с необходимостта от *проучването на пазара*, дали има реален потенциал да бъде успешно направен и пуснат в действие. Реализацията на софтуер трябва да удовлетворява както изискванията на собственика на продукта така и изискванията на крайните потребители, като задоволява техните потребности. Бързо променящите се технологии налагат още повече да се направи добро *проучване на пазара* с цел да се избере възможно най-добрият подход за реализация, дългосрочно функциониране, бъдещо подобрение и поддръжка на продукта.

Тук пишете какво мислите за вашия продукт и какво очаквате от него: какъв е вашият продукт; как очаквате да реагират потребителите; ще имат ли интерес от него; има ли подобни продукти; как очаквате вашия продукт да се впише сред останалите; ще има ли според вас успех и защо очаквате такъв; как очаквате да бъдете конкурентни с вашия продукт на вече доказали се подобни софтуерни продукти; с какви технологии са реализирани продукти на ваши конкуренти и др.

1.2. Преглед на технологии и езици за програмиране

Анализ на всичко важно свързано с уеб сайтовете, разработките с трислойните модели и т.н. Описвате езици за програмиране и база данни. Тази точка е свързана със съществуващите технологии, които могат да се използват при създаването на уеб приложения. Задължително трябва да бъдат разгледани технологиите, които ще използвате в проекта.

1.3. Избор на технологии и езици за програмиране

Информация относно избраните технологии. Обосновете се защо точно тях ще използвате. Акцента трябва да бъде, че те са съвременни, гъвкави, ефективни, гарантират бъдещо разширяване и поддръжка. Вижте например как рекламират Visual

Studio, какво казват за MS SQL Server, езика C#. Тук напишете всичко което ще използвате във вашия проект като технологии и програмни езици. За всяка една от тях напишете нещо положително.

2.ВТОРА ГЛАВА – ПРОЕКТИРАНЕ

Описание и анализ на изискванията към програмния продукт; разглеждане на потребителски истории и случаи на употреба на продукта; прототип на потребителски интерфейс на приложението. Избор на подход за изграждане на приложението и проектиране на класовете или на таблиците от БД./8-11 стр./

Проектирането (планирането) е част от всеки проект . В процеса на проектиране се откриват конкретни задачи свързани със самия проект. Важна част от създаването на дипломния проект е определяне на изискванията и техния анализ. За да се създаде уеб приложение ще се премине през различни етапи на проектиране. Реализацията на уеб приложението ще бъде направено в следващата глава, а тестването ще се извършва по време на разработка.

2.1. Описание на изискванията

2.1.1. Общо описание на изискванията

Изискванията са свързани с процеса на определяне на услугите, които клиентът изисква от системата (уеб приложението) и на ограниченията, при които тя работи и са разработва. Самите изисквания са описания на системните услуги и на ограниченията на системата. Изискванията могат да варират от много абстрактни описания на дадена системна функционалност или ограничения до много точни и детайлни математически функционални спецификации.

Проблеми при изискванията могат да възникнат:

- Поради неточно описание на изискванията.
- Двусмисленост - изисквания могат да бъдат интерпретирани по различен начин от разработчиците и потребителите. Например, изискването „подходящата среда“ за потребителя може да означава, среда подходяща за отваряне на документ. За разработчиците „подходящата среда“ може да означава среда в която да разработят успешно своя проект.

Типове изисквания:

- Системни изисквания – представляват структурирана документация с описание на функциите, услугите и работните ограничения на системата. Определят какво трябва да се реализира и може да бъде част от договора между клиента и разработчика

- Потребителски изисквания – Указания на естествен език и диаграми на услугите, които системата ще предоставя, както и съответните оперативни ограничения.

2.1.2. Функционални изисквания

Функционалните изисквания определят, какво трябва да прави системата. В нашият случай системата представлява уеб приложение. Функционалността на системата определя изхода спрямо нейния вход. Функционални изисквания представляват описание на услугите, които системата трябва да предостави, начинът, по който трябва да реагира на конкретни входни данни и нейното поведение в конкретни ситуации.

Функционалните изисквания включват:

- Описание на функции или услуги на системата.
- Зависят от типа на софтуера, потенциалните потребители и типа на системата, в която ще бъде използван софтуера.
- Функционалните потребителски изисквания могат да бъдат описания на високо ниво на това какво трябва да прави системата, но функционални системните изисквания трябва да описват системната функционалност в детайли.

Бележка: Функционалните изисквания ще бъдат подробно разгледани в раздел „Анализ на функционалните изисквания“.

2.1.3. Нефункционални изисквания

Ограничения върху услугите или функционалността на системата като времеви ограничения, ограничения върху процеса на разработване, използваните стандарти и др. Нефункционалните изисквания могат да бъдат по-критични от функционалните и от тях може да зависи пригодността на системата.

Нефункционалните изисквания определят:

- Системни характеристики и ограничения като надеждност, време за отговор и др.
- Изисквания към хардуера, технологията за разработка, програмния език или методите на разработване.

Бележка: Нефункционалните изисквания ще бъдат подробно разгледани в раздел „Анализ на нефункционалните изисквания“.

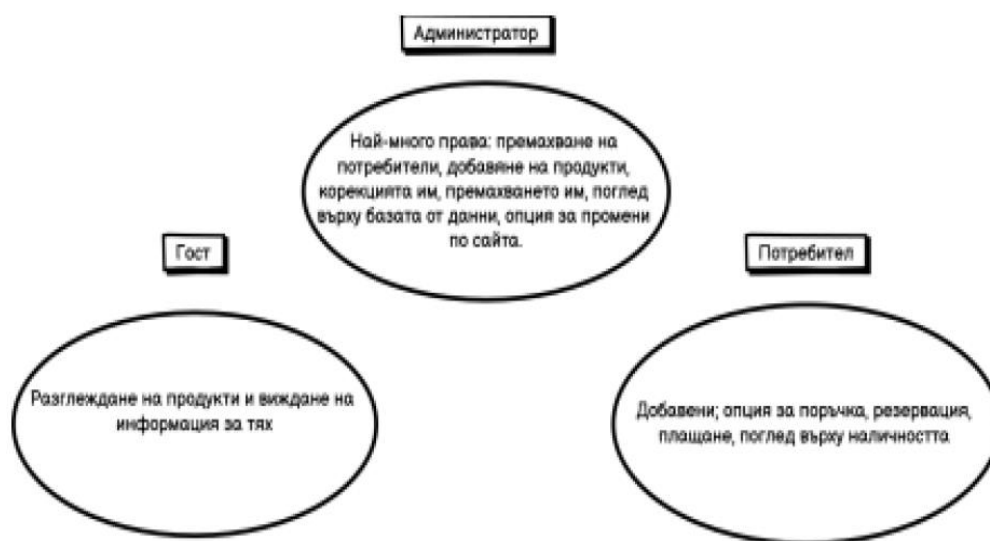
2.2. Анализ на функционалните изисквания

Фаза анализ се занимава с това какво трябва да върши приложението. В раздел проектиране и глава реализация ще бъде разгледан въпросът как точно приложението ще постигне набелязаните цели от фазата на анализа. Целта на анализа е да предостави ясна спецификация за потребителя. За разлика от фазата на проектиране и реализация, фазата на анализа не акцентира на техническите въпроси в проекта. Резултатът от анализа представлява документиране на функционалните изисквания за уеб приложението.

2.2.1. Изисквания към правата на потребителите

Функционалността на уеб приложението се определя в голяма степен и от това какви са правата на определена група потребители. В уеб приложението ще има **три или четири** групи потребители, както е показано на *фигура 1*.

Сложете ваша фигура, която показва правата на потребителите.



Фигура 1. Потребителски права в уеб приложението

Администратор – той има най-големи права при използването на приложението. Той може да въвежда данни за продукти, данни за потребители, да актуализира и изтрива данни от базата с данни. Това е потребителят с най-много правомощия в приложението. Той има по-голям набор от функции спрямо останалите потребители (промяна на информацията за продуктите - добавяне, премахване,

преглед на потребителите и т.н.). **Опишете правата на администратора в зависимост от контекста на дипломния ви проект.**

Служител – ако имате такъв случай, опишете нещо за него

Регистриран потребител – има достъп до определен набор от функции на приложението и може да използва услугите на сайта: разглеждане на продукти, поръчки на продукти, информация за негови поръчки, филтриране на продукти и др. Неговите права са по-големи от тези на гостите. **Опишете правата на регистрирания потребител в зависимост от контекста на дипломния ви проект.**

Гост – има ограничени права - може да разглежда част от сайта, може да се регистрира. Неговите права са по-малки от тези на регистрираните потребители.

2.2.2. Изисквания към функционалния дизайн

В тази точка опишете дизайна на страниците обаче пречупена през функционалността. Какви менюта ще има и каква информация ще бъде предоставена, за да изпълнят набелязаните цели, относно функционалността на сайта. Това което напишете тук ще бъде използвано в анализа по-долу за дефиниране случаите на употреба и потребителските истории.

Уеб приложението ще разполага със следните публично достъпни секции: какви са менютата. **Опишете какво ще има в другите страници, които водят към определяне на функционалността на сайта.**

Начална страница - с влизането в сайта потребителят ще види началната страница, която представя следната информация:

- преход към други страници, чрез избор на бутони от менюто
- новинарски бюлетин, рекламни постове, брошури и изображения
- друга актуална информация и материали отнасяща се към сайта - например, файл с каталог на продукти.

Функционалността на началната страница се изразява в това какъв избор може да направи потребителя на сайта, например, кликва на реклама, сваля файл, избира бутон от меню и др. Потребителят ще има възможност да избере бутони и други функционални елементи за управление на екраните и достъп до съдържанието в сайта.

Страница регистрация – ще предостави възможност за регистрация на потребителски акаунти. **Напишете още информация – каква информация от потребителя ще се изисква, какви са бутоните, които определят функционалността на страницата.**

Страница продукти - ще разполага с информация в реално време на всичко налично и всичко изчерпано което не е налично от продуктите. Обновяването е в реално време, за да има постоянно актуална информация за продуктите предлагани в сайта. Функционалността на страницата с продукти се изразява в това, че потребителят ще има възможност да кликне върху избран продукт и да види подробности за него. Когато потребителят се намира в тази страница менюто за преход към други страници остава достъпно **и т.н.**

Страница - опишете подобно на горните страници, като акцента трябва да бъде върху функционалността на страницата. Например в страница контакти можете да имате бутон, който да ви изпраща в Google Maps, или да имате бутон за изпращане на имейл и др.

2.2.3. Потребителски истории (user stories)

Потребителските истории се използват при разработването на софтуер, като начин да се помогне на разработчиците да разберат желанията и нуждите на своите потребители. Потребителските истории са кратки и следват прост шаблон за изразяване. Потребителските истории могат да се вграждат в епоси, които са по-широки твърдения описващи „голямата картина“ на потребителския опит.

Предимства:

- кратки
- разбираеми за потребителите и разработчиците
- Не е нужна поддръжка
- Не налагат особени усилия

Недостатъци

- Може да са непълни
- Различна интерпретация

Потребителските истории имат за цел да представят гледната точка на потребителя, а не гледната точка на създателя (разработчика на програмата). Потребителите не са експерти относно реализацията на програмата и нямат гледната точка на създателя. Потребителските истории могат да са от голяма полза при определяне на случаи на употреба.

Шаблон за потребителски истории (кой-какво-защо):

„Като (потребител) искам да (действие/възможност), за да (цел).”

Този шаблон изобразява представата на потребителя, като идентифицира какво иска и защо го иска.

Примери:

- „Като администратор искам да напиша ясна информация, за продуктите, за да спечеля доверието на клиентите.“
- „Като потребител искам да видя добра информация, за продуктите, за да зная какво купувам.“
- „Като потребител искам да мога да изтегля каталог, за продуктите, за да разгледам продуктите в по удобен вид.“
- „Като администратор искам да създам отчет, за да виждам информация за клиенти изтеглили каталога за продуктите.“
- „Като администратор искам да създам реклама за продукти, която се показва на клиенти изтеглили каталога, за да спечеля тяхното внимание.“
- „Като администратор мога да осъществя достъп до контролния панел, за да управлявам сайта.“
- „Като потребител мога да минимизирам информацията за продуктите, за да видя повече продукти на екрана.“

Напишете потребителски истории, в зависимост от контекста на вашия дипломен проект. Опитайте се да напишете такива истории, които наистина са реализирани в сайта. Помислете за потребителски истории, които след това да бъдат превърнати в случаи на употреба.

Няма доказателства, че използването на потребителски истории повишава успеха на софтуера или производителността на разработчиците. Въпреки това, потребителските истории улесняват проектирането на софтуерния продукт.

2.2.4. Случаи на употреба (use cases)

Случаите на употреба представя множество от възможности при използване на уеб приложението. Тези възможности са свързани с начина, по който сайта се използва от различни потребители. Анализирането на случаите на употреба дефинират отговори на различни въпроси, като „Какво трябва да направим в даден случай?“, „Какви трябва да бъдат правата на определен тип потребителите?“ и др.

Това е рамка за документиране, която съответства на случаи на избор и решения, подобно на оператори *if...then... else*, които помагат на програмистите да обмислят

проблемите и да предоставят решението в стандартното програмиране. Определянето на случаите на употреба, за уеб приложението, ще се направи според информацията представена в „Изисквания към функционалния дизайн“ и „Потребителски истории“.

Общ вид на карта за определяне на случай на употреба

Основно действащо лице	Гост, регистриран потребител, служител , администратор.
Предварителни условия	Условията които трябва да са изпълнени за да се изпълни случая на употреба.
Основен успешен сценарии	Описание за начина по който уеб приложението работи.
Алтернативен сценарии	Описание за начина по който уеб приложението работи при алтернативен начин на употреба.
Специални изисквания	Изисквания които не влизат в Основен успешен сценарии и Алтернативен сценарии .
Неразрешими проблеми	Въпроси които трябва да получат отговор, преди случаят на употреба да бъде завършен.

Стъпки за оформяне на случаи на употреба

- Определя се кои са типовете потребители.
- Определя се целта, която всеки потребител трябва да постигне.
- Попълват се всички критерии посочени в картата.
- Прави се оценка на случаите на употреба, преди да се премине към реализация в уеб приложението.

Създаване на карти за случаи на употреба за уеб приложението - примери

Карта 1

Основно действащо лице	Клиент
Предварителни условия	Клиентът вижда формата за попълване на адрес и бутон за изпращане на данните.
Основен успешен сценарии	Клиентът попълва формата, изпраща данните с натискане на бутона. Системата приема успешно попълнените данни във формата и извежда съобщение за приети данни.
Алтернативен сценарии	Системата открива грешни данни в попълнената форма и показва отново формата със съобщение за грешка.
Специални изисквания	Формата и бутоните да бъдат удобни с размери, които позволяват на хора със слабо зрение да попълнят формата.
Неразрешими проблеми	Ще се използват ли форми, които позволяват на потребителя да избира от списък с опции (например град, област и т.н.) или той ще трябва на ръка да попълва всички полета.

Направете карти за случаи на употреба отнасящи се до вашия сайт

2.2.5. UML диаграми за случаи на употреба

Функционалният модел позволява изготвянето на UML диаграми за различни случаи на употреба. Тези диаграми не винаги са достатъчно детайлни, за да бъдат използвани отделно от текстовите описания. Въпреки това те са добро допълнение, което осигурява визуализация на случаите на употреба и действащите лица в системата.

UML диаграми по създадените карти:

- На *фигура 2* е показана диаграма на функционален модел за случаи на употреба по **Карта ?**
- На *фигура 3* е показана диаграма на функционален модел за случаи на употреба по **Карта ?**
- На *фигура 4* е показана диаграма на функционален модел за случаи на употреба по **Карта ?**



Фигура 2. UML диаграма на случай на употреба по **Карта ?**

Начертайте схеми подобни на показаната в документа

2.3. Анализ на нефункционалните изисквания

Нефункционални изисквания биват:

- Производителност
- Сигурност
- Качество на софтуера
- Бизнес правила

Изисквания към дизайна

Опишете изисквания за външния вид на сайта – подреждане на елементи в страниците, използвани шрифтове, цветове, контраст, анимации, какви видове контроли ще се използват и с каква големина, разположение на контролите и др.

Изисквания към софтуера за разработка

- Visual Studio Community 2019
- Библиотеки необходими за проекта: ASP.NET Core Identity; напишете и други
- MS SQL Server

Изисквания към технологиите и езиците за разработка

- Базов проект за ASP.NET Core включващ MVC (Controller-View-Model)
- C#, SQL, HTML, CSS, Java script и др.

Изисквания към хардуера за разработка

Използвайте изискванията за минимална система посочени в документацията на софтуера, който ще използвате.

Изисквания към хостинга на уеб приложението

Уеб хостът, който ще се използва за работа на приложението в Интернет, трябва да поддържа използваните от уеб приложението технологии: .NET 5.0, MS SQL Server, MVC, LINQ, Mail Server и др..

Доставчикът на хостинг услуги трябва да:

- използва бързи дискове и кеширане на данни, за минимално време на отговор към клиента;
- поддържа подходящ трансфер на данни за уеб приложението;
- да осигури работа на уеб приложението, 24 часа и 7 дни в седмицата;
- осигури достатъчно дисковото пространство за уеб приложението и базата данни, както и за тяхното разрастване в бъдеще;
- позволява безпроблемно преминаване към друг хостинг план;
- поддържа добър и удобен панел за управление на услугите.

Бележка: С цел да не се прави реклама, тук не трябва да се посочват конкретен доставчици на хостинг услуги.

2.4. Проектиране

2.4.1. Представяне

Във фаза проектиране ще се използват функционалните изисквания от фаза анализ, за да се определят различните подсистеми и класове. Във фаза проектиране ще се извлекат отделни елементи, които ще бъдат описани с класове, ще се определят задачите за тези класове както и отношенията помежду им. Целта на тази фаза е да се изготви скица (техническа документация) на уеб приложението, което трябва да се създаде. Скицата включва всички подсистеми в това число и базата данни, класове и техните отношения, като предоставя ясно определяне на зависимостите и

взаимодействията между отделните системи и класове. Създадената скица ще предостави помощ и разбиране на процесите в етапа на разработка. Във фазата на проектиране няма да се определят техниките и инструментите, които ще се използват във фазата на реализация. Тъй като изграждането на базата данни е по модела **code first**, първо ще разгледаме необходимите класове, които трябва да създадем, а след това ще представим таблиците, които ще бъдат създадени в базата данни и връзката между тях.

2.4.1. Класове и взаимоотношения

Тук ще се опишат класовете, които вие създавате и тяхната взаимовръзки. В тази точка не описвайте класове, които се генерират автоматично от Visual Studio.

За реализацията на класове и връзка между тях ще се използват резултатите от анализа направен за функционалните изисквания. Нефункционалните изисквания определят използването за специфичен избор на технологии, с които веб приложението да бъде реализирано и тази част ще бъде разгледана в следващата глава.

В раздела на функционалните изисквания разгледахме функционалността на веб приложението за отделните страници, направихме множество случаи на употреба и потребителски истории. От направения анализ се вижда, че се нуждаем от класове в следните страници:

***Начална страница** – използвайте дефинираните по-горе карти на случаи на употреба за тази страница, за да определите класовете, които изпълняват функционалността описана в картата.*

***Страница регистрация** - използвайте дефинираните по-горе карти на случаи на употреба за тази страница, за да определите класовете, които изпълняват функционалността описана в картата.*

***Страница продукти** - използвайте дефинираните по-горе карти на случаи на употреба за тази страница, за да определите класовете, които изпълняват функционалността описана в картата.*

***Страница** - използвайте дефинираните по-горе карти на случаи на употреба за тази страница, за да определите класовете, които изпълняват функционалността описана в картата.*

.....

2.4.2. Проектиране на база данни

Нефункционалните изисквания определят използването на *MS SQL Server* за управление на базата данни. За *функционалните изисквания* определянето на таблиците и връзката между тях (релациите) в базата данни са следствие от класове, които реализират случаите на употреба. Таблиците в базата данни, които автоматично се генерират в проекта от Visual Studio Community 2019 ще бъдат посочени в следващата глава, когато реално бъдат създадени.

Тук трябва да опишете класовете и техните полета, от които системата се нуждае, за да изпълни описаните по-горе функционални изисквания, както и връзките между отделните класове.

Пример:

Системата трябва да пази информация за регистрирани потребители, като за всеки от тях се съхраняват данни поне за:

- име
- фамилия
- имейл
- телефон
- адрес

За всяка туристическа забележителност се съхранява информация минимум за:

- име на забележителност
- описание
- снимка
- категория

За всяка категория на туристическа забележителност се съхранява информация минимум за:

- име на категория /паметник, музей, природна забележителност и др./

За всяка екскурзия до туристическа забележителност се съхранява информация минимум за:

- начална дата и час
- крайна дата и час
- описание
- снимка

- забележителност
- максимален брой посетители
- цена за един посетител

За всяка резервация се съхранява информация минимум за:

- дата и час
- екскурзия
- брой места
- потребител

Може да използвате и прототип на диаграма на базата от данни.

2.5. Прототипи на потребителския интерфейс

Прототипите дават нагледа представа за визията на софтуерния продукт. Те позволяват на дизайнерите да покажат продукта си, което го прави по-лесен за разбиране. Прототипи могат да се създават по време на всеки етап от процеса по дизайн, за да помогнат да се демонстрират идеи, които трудно биха се изразили само с думи.

Тук направете ваши прототипи на страниците. Прототипът да бъде направен с помощта на програма, например Balsamiq Wireframes for Desktop: <https://balsamiq.com/wireframes/desktop/#>.

Пример:



Фигура ?.....?

3. ТРЕТА ГЛАВА – РЕАЛИЗАЦИЯ

Същинската част на дипломния проект, която е с най-голям обем. Да включва структура на базата от данни /чрез E/R диаграма/ и описание на полетата на всяка таблица, описание на архитектурата на проекта, контролери, изгледи, сървиси. Описание на реализираните функционалности, в това число чрез представяне и описване на екрани от приложението. /15-18 стр./

3.1. Създаване на проект

За реализация на сайта ще се използва базов проект за ASP.NET Core на езика C#. Базовият проект има вграден MVC модел, който ще бъде разширен за да се изпълнят целите на уеб приложението.

Стъпки за изграждане на уеб приложението

Опишете стъпките /последователността/ за изграждане на приложението.

3.2. Структура на базата от данни

Описание на полетата за всяка таблица от базата данни. Връзки между таблиците. E/R диаграма на базата данни.

3.3. Изграждане на уеб приложението

3.3.1. Архитектура на приложението

Описвате архитектурата на уеб приложението. Може да добавите схема на използваните класове /на структурата на приложението./

3.3.2. Контролери

Описвате контролерите и тяхната функционалност

3.3.3. Изгледи

Описвате изгледите и тяхната функционалност. Задължително слагате снимка за всеки изглед.

3.3.4. Сървиси

Описвате сървисите и тяхната функционалност

3.3.5. Други технологии и функционалности

ЗАКЛЮЧЕНИЕ

Обобщава постигнатото в дипломния проект, представя възможностите за доразвиване и усъвършенстване, както и възможностите за приложение на създадения софтуер. /1-2 стр./

ИЗПОЛЗВАНА ЛИТЕРАТУРА

Включва цитираната и използвана в записката на дипломния проект литература. Започва на отделна страница от основния текст. При имената на авторите първо се изписва фамилията. Всички описания в списъка с използваните източници трябва да са подредени по азбучен ред според фамилията на първия автор на всяка публикация.

1. Наков, С. и колектив. Принципи на програмирането със C#. Фабер, 2018
2. Наков, С. и колектив. Програмиране за .NET Framework. БАРС, София, 2006
3. Свободно учебно съдържание по програмиране и ИТ за българските училища на Работна група „Образование по програмиране и ИТ“ – <https://github.com/BG-IT-Edu>
4. <https://docs.microsoft.com/en-us/documentation/>
5. <https://www.w3schools.com/>

ПРИЛОЖЕНИЕ

Всяко на нова страница и да са номерирани /Приложение 1, Приложение 2 .../
Приложенията съдържат документация, която не е намерила място в текста поради ограниченията в обема ѝ или за по-добра прегледност/подредба. В текста трябва да има препратка към всяко приложение.