**ДИПЛОМЕН ПРОЕКТ**

**ЗА ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ**

по професия код 481030 „Приложен програмист“

специалност код 4810301 Приложно програмиране“

ТЕМА: „ПРИЛОЖЕНИЕ ЗА УПРАВЛЕНИЕ НА КНИГОХРАНИЛИЩЕ“

Автор:

Александър Иванов Баев, клас XII В

Ръководител:

Име, Фамилия

БургасСЪДЪРЖАНИЕ

[1 Увод 3](#_Toc155035766)

[2 Цели и обхват на софтуерното приложение 3](#_Toc155035767)

[3 Анализ на решението 4](#_Toc155035768)

[3.1 Потребителски изисквания и работен процес 4](#_Toc155035769)

[3.2 Примерен потребителски интерфейс 5](#_Toc155035770)

[3.3 Диаграми на анализа 5](#_Toc155035771)

[3.4 Модел на съдържанието / данните 5](#_Toc155035772)

[4 Дизайн 6](#_Toc155035773)

[4.1 Реализация на архитектурата на приложението 6](#_Toc155035774)

[4.2 Описание на слоевете, предназначението им, библиотеки и методи включени в съответния слой. 6](#_Toc155035775)

[4.3 Организация и код на заявките към база от данни 6](#_Toc155035776)

[4.4 Наличие на потребителски интерфейс (конзолен, графичен, уеб) 6](#_Toc155035777)

[5 Ефективност и бързодействие на решението 7](#_Toc155035778)

[6 Тестване 7](#_Toc155035779)

[7 Заключение и възможно бъдещо развитие 7](#_Toc155035780)

[8 Използвани литературни източници и Уеб сайтове 7](#_Toc155035781)

[9 Приложения 8](#_Toc155035782)

[10 Критерии и показатели за оценяване 9](#_Toc155035783)

# Увод

В днешния дигитален свят ефективното управление на книжните ресурси и библиотечните системи е от съществено значение. Традиционните методи за проследяване и управление на инвентара на книги често са тромави, изискват ръчен труд и са податливи на грешки. Това затруднява работата на администраторите и ограничава достъпа на читателите до необходимите ресурси. Освен това, липсата на автоматизирана система за управление на книжния фонд води до затруднения при организирането на наличните ресурси, следенето на наемите и поддръжката на актуална база данни.

Настоящият проект има за цел да създаде уеб базирано приложение за управление на книгохранилище, което автоматизира и улеснява процесите по управление на инвентара, отдаване под наем на книги и следене на наличността им. Чрез внедряването на модерни технологии се осигурява по-добра организация, подобрена ефективност и по-лесен достъп до книжните ресурси. Системата позволява на администраторите да регистрират нови книги, да управляват информацията за тях, както и да проследяват кои книги са налични и кои са в процес на наемане. Освен това, приложението предоставя възможност за генериране на отчети, което подпомага анализирането на използваемостта на ресурсите и улеснява вземането на управленски решения.

Важна част от разработката на този проект е използването на подходящи софтуерни технологии и архитектури, които гарантират стабилност, мащабируемост и лесна поддръжка на системата. За моделирането на решението се използва езикът UML, като различните диаграми (диаграма на случаите на употреба, диаграма на класовете и диаграма на дейностите) се създават в средата за моделиране Lucidchart. Тези диаграми илюстрират логическата структура на системата, взаимодействието между потребителите и различните компоненти на приложението, както и основните процеси, които системата поддържа.

Проектът се основава на многослойна архитектура, което осигурява ясно разделение между различните компоненти на приложението – база данни, бизнес логика и потребителски интерфейс. В технологичен аспект се използва ASP.NET Core за сървърната част, Entity Framework Core за работа с базата данни и MVC/Razor Pages за фронтенд слоя. Това позволява създаването на гъвкаво и мащабируемо решение, което може да се разширява в бъдеще с нови функционалности и интеграции.

Останалата част от документацията е структурирана, както следва:

* **Цели и обхват на софтуерното приложение**
  + Описва целите и обхвата на приложението, като дефинира основните функционалности, които се реализират.
* **Анализ на решението**
  + **П**редставя анализа на решението, включително потребителските изисквания, работния процес и логическата структура на данните.
* **Дизайн**
  + Разглежда архитектурата и дизайна на приложението, като обяснява как различните компоненти са организирани и как взаимодействат помежду си.
* **Ефективност и бързодействие на решението**
  + **А**нализира ефективността и производителността на решението, като разглежда начини за оптимизация и подобрение на работата на системата.
* **Тестване**
  + **Точката** е посветена на тестването на системата, включително стратегия за извършване на Unit тестове и интеграционни тестове.
* **Заключение и възможно бъдещо развитие**
  + Разглежда възможности за бъдещо развитие и разширяване на функционалността на приложението.
* **Използвани литературни източници и Уеб сайтове**
  + Съдържа използваните литературни източници и уеб сайтове, които служат като основа за изследването и разработката.
* **Приложения**
  + Включва приложения с допълнителни материали, като детайлни диаграми и примери за код.
* **Критерии и показатели за оценяване**
  + Дефинира критериите и показателите за оценяване на проекта, които се използват за неговото финално представяне.

# Цели и обхват на софтуерното приложение

Софтуерното приложение за управление на книгохранилище е създадено с цел да автоматизира процесите, свързани с управлението на книги, наематели и наемите на книги. Чрез интеграцията на съвременни технологии, приложението предоставя ефективен начин за администриране на библиотечния инвентар, управление на потребителите и генериране на отчети.

**Обхват на софтуерното приложение**

Приложението е предназначено за администратори, които ще управляват книжните ресурси и процесите, свързани с тях. Те ще имат достъп до следните функционалности:

* **Автентикация и сигурност**
  + Вход в системата (login)
  + Възстановяване на забравена парола (forgot password)
  + Нулиране на парола (reset password)
  + Промяна на паролата от профила на администратора (change password)
* **Управление на книги**
  + Добавяне на нови книги в системата
  + Редактиране на информация за вече въведени книги
  + Преглед на детайлите на дадена книга
  + Изтриване на книги от каталога
* **Управление на наематели**
  + Регистриране на нови наематели
  + Поддържане на информация за съществуващи наематели
  + Връзка с потребителите
* **Управление на наеми**
  + Добавяне на нови наеми
  + Проследяване на статуса на даден наем (активен, просрочен, върнат)
* **Генериране на отчети**
  + Преглед на наличните книги
  + Анализ на най-често наеманите книги
  + Анализ на най-активните наематели
  + Преглед на върнатите наеми

Приложението осигурява интуитивен потребителски интерфейс, който улеснява достъпа до информация и минимизира времето, необходимо за изпълнение на административни задачи.

**Цели на софтуерното приложение**

Основната цел на разработката е да създаде ефективно, сигурно и лесно за използване уеб приложение, което да подпомага управлението на книгохранилището. Това се постига чрез:

* Автоматизиране на ръчните процеси по администриране на книжните ресурси и наемите
* Подобряване на контрола върху наличността на книгите
* Улесняване на администраторите при поддръжката на базата с наематели
* Осигуряване на механизъм за генериране на полезни отчети и анализ на данните
* Гарантиране на сигурност при достъпа до системата чрез надеждни механизми за автентикация и управление на потребителски профили

Разработеното приложение ще оптимизира работата на книгохранилищата, ще намали риска от грешки и ще повиши ефективността на администраторите при управлението на библиотечните ресурси.

# Анализ на решението

## Потребителски изисквания и работен процес

Тук опишете най-общо работния процес като вход, обработка и изход, тоест:

* какво представлява входното съдържание/данни и откъде и как се получава
* **как ще се обработва и запазва в системата
* какво трябва да се получи като изход и къде и как ще се използва.

За целта използвате диаграми на случаи на употреба /с потоци от събития/ и диаграми на дейностите. Структурирайте диаграмите по подходящ начин – напр. по нива на абстракция или като съставни диаграми с връзки към други диаграми.

Забележки:

1. Всички диаграми трябва да са създадени в средата dwaw.io, оригиналните файлове трябва да са добавени в репозиторито на проекта.
2. Диаграмите спазват конвенциите за описание на UML стандарта и са добавени в документацията към проекта.

## 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

Тук опишете резултата от анализа на проблема с UML диаграми

* ER диаграма на базата данни
* използване на клас диаграми на анализа /с класове със стереотипи/ За по-сложните контролни класове представете диаграми на състоянието /евентуално йерархични/
* диаграми на последователността и на комуникацията

Забележки:

1. В зависимост от спецификата на проекта трябва да бъдат разработени съответно необходимите диаграми.

## 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

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

## Организация и код на заявките към база от данни

Описание на инструментариума за достъп до базата данни от гледна точка на програмния код. Описание на методите за извличане, добавяне и изтриване на обекти в базата данни.

## 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

Съдържа описание и анализ на известните решения, като се цитират съответните литературни източници.

# 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

В заключение, обобщете резултатите от работата ви по проекта, както и предимствата и ограничеността на използваните технологии / езици / методи. Укажете какви алтернативи могат да се използват и техните предимства и недостатъци. Опишете каква е използваемостта на подобни решения в практиката и какво бихте предложили като насоки за бъдещо развитие на вашето решение.

# 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.