**ПРЕДМЕТ: РАЗРАБОТКА НА СОФТУЕР**

**ДЕЙНОСТ: УЧЕНИЧЕСКИ ПРАКТИКИ 2**

***2022-2023***

**Тема: „Система за управление на хранителен магазин“**

**КУРСОВ ПРОЕКТ**

*Автор:*

*Поля Димитрова Димитрова, клас XI В*

Бургас

Съдържание

[1. Въведение 4](#_Toc136207486)

[2. Цели и обхват на софтуерното приложение 4](#_Toc136207487)

[3. Анализ на решението 4](#_Toc136207488)

[3.1 Потребителски изисквания и работен процес 4](#_Toc136207489)

[3.2 Примерен потребителски интерфейс 5](#_Toc136207490)

[3.3 Диаграми на анализа 5](#_Toc136207491)

[3.4 Модел на съдържанието 6](#_Toc136207492)

[3.4.1 Входни данни 6](#_Toc136207493)

[3.4.2 Обработка на данни 6](#_Toc136207494)

[3.4.3 Изходни данни 7](#_Toc136207495)

[4 Дизайн 7](#_Toc136207496)

[4.1 Реализация на структурата 7](#_Toc136207497)

[4.1.1 Потребителски интерфейс (Presentation Layer): 8](#_Toc136207498)

[4.1.2 Бизнес логика (Business Logic Layer): 8](#_Toc136207499)

[4.1.3 Съхранение на данни (Data Access Layer): 9](#_Toc136207500)

[4.2 Организация и код на заявките към база от данни 9](#_Toc136207501)

[4.2.1 Извличане на обекти (Querying): 9](#_Toc136207502)

[4.2.2 Добавяне на обекти (Adding): 10](#_Toc136207503)

[4.3 Наличие и интуитивност на потребителски интерфейс (конзолен, графичен, уеб) 10](#_Toc136207504)

[4.3.1 Регистрация и вход: 10](#_Toc136207505)

[4.3.2 Меню за навигация: 10](#_Toc136207506)

[4.3.3 Визуализация на данни: 10](#_Toc136207507)

[4.3.4 Въвеждане и редактиране на данни: 11](#_Toc136207508)

[4.3.5 Обработка на заявки и операции: 11](#_Toc136207509)

[5 Тестване 11](#_Toc136207510)

[5.1 Подходи за тестване: 11](#_Toc136207511)

[5.2 Инструменти за тестване: 11](#_Toc136207512)

[5.3 Ръчно тестване: 12](#_Toc136207513)

[5.4 Отстраняване на грешки: 12](#_Toc136207514)

[6 Заключение и възможно бъдещо развитие 12](#_Toc136207515)

[6.1 Заключение: 12](#_Toc136207516)

[6.2 Възможно бъдещо развитие: 12](#_Toc136207517)

[7 Използвани източници и Уеб сайтове 13](#_Toc136207518)

# Въведение

Настоящият документ представлява и представя цялата информация за проекта на Поля Димитрова от 11В клас – „Ученически практики 2“.

# Цели и обхват на софтуерното приложение

Проектът представлява система за менажиране на хранителен магазин в по-големите вериги. В нея ще могат да се регистрират служители и клиенти, получавайки определени правила за достъп и използване на приложението. Целта е осигуряване на по-бърз и лесен начин за намиране и достъпване на продукти в магазин.

# Анализ на решението

## Потребителски изисквания и работен процес

Системата за достъп до приложението се нуждае от направен и регистриран успешно акаунт, чрез който да може потребителя да влезе и използва услугите на апликацията. Данните, които ще потрябват за направата на акаунт и неговото осигурение са:

* Имейл, потребителско име и парола
* При неправилен или несъществуващ имейл на екрана ще излезе надпис за грешка
* При вече заето потребителско име на екрана ще се изведе надпис за грешка
* При неправилна парола потребителя ще има изборът да опита отново или да избере „Забравена парола“ след което чрез имейл ще бъде доставена нова форма за презаписването на новата парола в акаунта.

При успешно влизане в акаунта в зависимост от неговата роля потребителя ще може да достъпва различен вид услуги и правила за достъп в апликацията.

## Graphical user interface, application Description automatically generatedПримерен потребителски интерфейс

## 3.3 Диаграми на анализа

* [Class Diagram](https://lucid.app/lucidchart/cad4e5d5-845b-4de7-84e8-ee15874dedc1/edit?viewport_loc=-182%2C-1207%2C3488%2C3702%2C0_0&invitationId=inv_93b2c470-61c9-4915-9e8a-d38ed6a91c6b)

Клас диаграмите са едни от най-полезните и използвани типове диаграми в UML, тъй като те ясно очертават структурата на определена система чрез моделиране на нейните класове, атрибути, операции и връзки между обектите. В случай на моя проект диаграмата показва връзката между потребителя, клиента, разрешенията за достъп, различните видове роли, продуктите и тяхната наличност, магазина и законопроекта.

* [Use Case Diagram](https://lucid.app/lucidchart/cad4e5d5-845b-4de7-84e8-ee15874dedc1/edit?viewport_loc=-182%2C-1207%2C3488%2C3702%2C0_0&invitationId=inv_93b2c470-61c9-4915-9e8a-d38ed6a91c6b)

Целта на този тип диаграма използвана доста често в UML e да демонстрира различните начини, по които потребителят може да взаимодейства със системата. В случай на моя проект тя показва достъпът и разрешенията на мениджъра, служителя в магазинът и доставчикът. Тримата заедно имат няколко еднакви задължителни права в мрежата и това са влизане и излизане от системата, обновяване на профил и промяна на парола на техният акаунт. Всички останали позволения са на базата кой каква роля заема в приложението.

* [ER Diagram](https://lucid.app/lucidchart/4a15b81b-27b8-41b5-8007-174885acf888/edit?viewport_loc=-182%2C-1252%2C3488%2C3702%2C0_0&invitationId=inv_21674161-b4a6-40df-a942-b44f6177af8f)

Диаграмата на връзката между обекти (ER) e вид блок-схема, която илюстрира как „субекти“, като хора, обекти или концепции се свързват един с друг в рамките на една система. Те най-често се използват за проектиране или отстраняване на грешки в релационни бази данни в областта на софтуерното инженерство, бизнес информационните системи, образованието и научните изследвания. В случай на моя проект диаграмата показва непосредствената връзка между потребителя, клиента, разрешенията за достъп, различните видове роли, продуктите и тяхната наличност, магазина и законопроекта.

## Модел на съдържанието

В раздела "Модел на съдържанието" ще бъдат описани входни и изходни данни, както и процеса на обработка на данните в системата за менажиране на хранителен магазин. Това ще помогне да се разбере как информацията се предава, преобразува и използва в системата.

### Входни данни

В системата за менажиране на хранителен магазин, входните данни могат да бъдат разделени на следните категории:

* + Регистрационни данни на служители: Име, фамилия, потребителско име, парола, роля (например администратор, касиер и т.н.), контактна информация и други допълнителни данни за служителите на магазина.
  + Регистрационни данни на клиенти: Име, фамилия, потребителско име, парола, адрес, контактна информация и други допълнителни данни за клиентите на магазина.
  + Информация за продуктите: Име на продукта, категория, цена, наличност и други характеристики на продуктите, които ще бъдат предлагани в магазина.

### Обработка на данни

След като входните данни бъдат приети от системата, те ще бъдат обработени по следния начин:

* Регистрация на служители: Въведените данни за служителите ще бъдат проверени за валидност и запазени в база данни или друго хранилище на данни. Създава се потребителски профил за служителите, който включва достъпни права и роли според зададените правила за достъп.
* Регистрация на клиенти: Подадените данни за клиентите ще бъдат валидирани и съхранени в база данни или друго хранилище на данни. Създава се потребителски профил за клиентите, което им позволява да използват приложението и да имат достъп до функционалностите на системата.
* Управление на продукти: Информацията за продуктите ще бъде обработена и запазена в база данни или друго хранилище на данни. Това включва създаване, редактиране и изтриване на продукти, както и актуализиране на наличността на продуктите.

### Изходни данни

Системата за менажиране на хранителен магазин генерира следните изходни данни:

* Потвърждения и съобщения: Системата изпраща потвърждения и съобщения до служителите и клиентите във връзка с регистрацията, промените в данните, статуса на поръчки и други релевантни съобщения.
* Списъци с продукти: Системата предоставя списъци с продуктите в магазина, които могат да бъдат филтрирани по категория, наличност или други характеристики.
* Данни за клиенти и служители: Информацията за клиентите и служителите може да бъде извлечена от системата за нуждите на отчетността, анализа на данните и други административни цели.

# Дизайн

## 4.1 Реализация на структурата

Разделът "Реализация на структурата" описва основните компоненти и слоеве на системата за менажиране на хранителен магазин и тяхната функционалност.

### 4.1.1 Потребителски интерфейс (Presentation Layer):

Потребителският интерфейс е отговорен за представянето на данните и взаимодействието с потребителите на системата. Тук можете да включите следните аспекти:

* Дизайн на потребителски интерфейс: Описва визуалните елементи, макета и навигацията в приложението. Включва форми, таблиците, бутоните, менютата и други елементи на потребителския интерфейс.
* Интерактивност: Описва начина, по който потребителите могат да взаимодействат с приложението, включително въвеждане на данни, избор на опции, извеждане на информация и други видове взаимодействие.
* Валидация на данни: Описва правилата и методите за валидиране на входните данни, за да се гарантира, че потребителите въвеждат коректна информация.
* Обратна връзка и уведомления: Описва начините за предоставяне на обратна връзка и уведомления към потребителите, включително съобщения за грешки, успех или състояние на операциите.

### 4.1.2 Бизнес логика (Business Logic Layer):

Бизнес логиката съдържа основните правила и процеси, които управляват функционалността на системата. Включва следните аспекти:

* Управление на потребителски профили: Описва логиката за регистрация, вход и управление на потребителските профили за служители и клиенти. Това включва проверка на идентификационни данни, управление на роли и права за достъп и други свързани операции.
* Управление на продукти: Описва логиката за добавяне, редактиране, изтриване и търсене на продукти. Включва валидация на данните за продуктите, проверка на наличността, обработка на запитвания за търсене и други операции, свързани с продуктите в магазина.
* Управление на поръчки: Описва логиката за създаване, обработка и доставка на поръчките. Това включва валидация на данните за поръчките, свързване с потребителските профили, обработка на плащания и други операции, свързани с поръчките.

### 4.1.3 Съхранение на данни (Data Access Layer):

Слойът за съхранение на данни е отговорен за взаимодействието с базата данни или друго хранилище на данни. Тук можете да включите следните аспекти:

* Модели на данните: Описва структурата на данните, които ще бъдат съхранявани в базата данни. Включва създаване на модели за служители, клиенти, продукти, поръчки и други съответни данни.
* Връзка с базата данни: Описва начините за свързване с базата данни, включително конфигурацията на връзката, установяването на връзка и обработката на заявките към базата данни.
* Заявки към базата данни: Описва начините за изпращане на заявки към базата данни за извличане, въвеждане, редактиране или изтриване на данни. Включва използване на SQL или ORM (Object-Relational Mapping) инструменти за изпълнение на заявки и манипулация с данните.

## 4.2 Организация и код на заявките към база от данни

### 4.2.1 Извличане на обекти (Querying):

За извличане на данни от базата данни използвам следните методи върху DbSet (таблица) в контекста на базата данни:

* FirstOrDefault: Този метод извлича първия обект от заявката или връща null, ако няма открити обекти.
* SingleOrDefault: Този метод извлича единичния обект от заявката или връща null, ако няма открит обект или връща грешка, ако има повече от един обект.
* ToList: Този метод извлича всички обекти от заявката и ги връща като списък.

Тези методи позволяват да формирам заявки, които да извлекат определени обекти, отговарящи на зададени условия.

Резултатът от заявката може да бъде единичен обект, който отговаря на условията, или списък от обекти, в зависимост от използвания метод.

### 4.2.2 Добавяне на обекти (Adding):

За добавяне на нови обекти в базата данни използвам следния подход:

* Използвам метода Add върху DbSet (таблица) в контекста на базата данни. Този метод позволява добавяне на нови обекти в таблицата.
* Подготвям новите обекти, като зададавам стойности на техните свойства, в съответствие с данните, които искам да запиша в базата данни.
* След подготовката на обектите извиквам метода Add върху DbSet с подадените обекти като аргументи, за да ги добавя към таблицата.
* За да запиша промените в базата данни, извиквам метода SaveChanges върху контекста на базата данни.

След извикването на метода SaveChanges, новите обекти са добавени в базата данни и промените са запазени.

## 4.3 Наличие и интуитивност на потребителски интерфейс (конзолен, графичен, уеб)

### 4.3.1 Регистрация и вход:

* Потребителите имат възможност да се регистрират в системата, като предоставят необходимата информация и създават акаунт.
* След успешна регистрация, потребителите могат да влизат в системата, въвеждайки своите потребителско име и парола.

### 4.3.2 Меню за навигация:

* Потребителският интерфейс предоставя лесен и интуитивен начин за навигация в приложението.
* Потребителите имат възможност да избират различни опции и функционалности от предоставените менюта.

### 4.3.3 Визуализация на данни:

* Системата предоставя визуализация на различни видове данни, свързани с потребителски профили, налични ресурси, резултати от заявки и други.
* Визуализацията е представена в удобен и разбираем формат, който позволява на потребителите да разбират и взаимодействат с информацията.

### 4.3.4 Въвеждане и редактиране на данни:

* Потребителският интерфейс предоставя възможност за въвеждане на нови данни и редактиране на съществуващи данни в системата.
* Потребителите могат да въвеждат текстови стойности, избират от предварително дефинирани списъци, въвеждат числови стойности и използват други методи за въвеждане и редактиране на данни.

### 4.3.5 Обработка на заявки и операции:

* Системата изпълнява заявки и операции, изисквани от потребителите.
* Това включва изпращане на заявки към базата данни, обработка на данни, изчисления и изпълнение на други операции, които отговарят на нуждите и целите на потребителите.

# Тестване

## 5.1 Подходи за тестване:

* В проекта ще се използва различни подходи за тестване, включително модулно тестване, функционално тестване и интеграционно тестване.
* Модулното тестване ще се фокусира върху тестването на отделни модули и компоненти на приложението, за да се уверим, че те работят правилно и изпълняват очакваната функционалност.
* Функционалното тестване ще проверява дали цялостната функционалност на приложението отговаря на спецификациите и изискванията.
* Интеграционното тестване ще се използва за проверка на взаимодействието между различните компоненти и модули на системата.

## 5.2 Инструменти за тестване:

* Ще се използват различни инструменти за тестване, като например unit testing frameworks и automation testing frameworks.
* Unit testing frameworks, като например NUnit или xUnit, ще бъдат използвани за модулното тестване на отделните компоненти и функции на приложението.
* Automation testing frameworks, като например Selenium или Appium, може да бъдат използвани за функционалното и интеграционното тестване, като автоматизират тестовите сценарии и проверките.

## 5.3 Ръчно тестване:

* Освен автоматизираното тестване, ще бъде извършвано и ръчно тестване.
* Ръчното тестване ще включва преглед и проверка на интерфейса, въвеждане на различни данни и изпълнение на действия от страна на потребителите.
* Това ще помогне за откриване на потенциални проблеми и грешки, които не могат да бъдат покрити от автоматизираните тестове.

## 5.4 Отстраняване на грешки:

* Ако по време на тестването се открият грешки или недостатъци, те ще бъдат отстранявани от разработчиците.
* След отстраняването на грешките ще се повтори тестването, за да се уверим, че проблемът е решен и функционалността работи коректно.
* Тестовете ще бъдат извършвани систематично и в различни етапи на разработката, за да се осигури висока степен на качество и стабилност на приложението.

# Заключение и възможно бъдещо развитие

## 6.1 Заключение:

Проектът за системата за менажиране на хранителен магазин представлява успешно разработена система, която позволява регистрация и управление на служители и клиенти, с определени правила за достъп и използване на приложението.

Целта на системата е осигуряването на по-бърз и лесен начин за намиране и достъпване на продукти в магазин.

Процесът на разработка включи анализ на изискванията, проектиране на структурата на системата, имплементация на компонентите и тестване на функционалността.

Използваните технологии и инструменти, както и структурата на проекта, допринесоха за успешното изпълнение на целите и изискванията на системата.

## 6.2 Възможно бъдещо развитие:

В бъдеще, системата за менажиране на хранителен магазин може да бъде разширена и подобрена с добавянето на нови функционалности и поддръжка на по-голям обем от данни.

Едно от възможните разширения може да бъде интеграцията със системи за управление на запасите и доставките, което ще позволи по-ефективно управление на наличностите и повишена точност при доставките.

Друго разширение може да бъде включването на система за онлайн поръчки и доставки, което ще разшири клиентската база и ще предостави удобство и леснота на потребителите при пазаруването.

# Използвани източници и Уеб сайтове

* "Entity Framework Core Documentation" - уеб сайт <docs.microsoft.com> .Официалната документация на Entity Framework Core, която беше използвана за справка при работата с ORM инструмента.
* "Microsoft SQL Server Documentation" - уеб сайт <docs.microsoft.com>. Документацията на Microsoft SQL Server, която предостави информация и примери за работа с базата данни.

# 8 Критерии и показатели за оценяване

|  |  |  |
| --- | --- | --- |
| **Показател** | **точки** | **срок** |
| 2. [Цели и обхват на софтуерното приложение](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Цели_и_обхват)  3.1 [Потребителски изисквания и работен процес](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Потребителски_изисквания_и) | 5  5 | 04.03.2022 |
| 3.2 [Примерен потребителски интерфейс](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Примерен_потребителски_интерфейс)  3.3 [Диаграми на анализа](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Диаграми_на_анализа)  3.4 [Модел на съдържанието/данните](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Модел_на_съдържанието) | 5  5  5 | 18.03.2022 |
| 4.1 [Реализация на структура на приложението (3-layer),  Разделение на кода според предназначението му.  Допълване на Class диаграми/3.3/](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Дизайн). | 10  10 | 16.04.2022 |
| 4.2 [Организация и код на заявките към база от данни](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Дизайн). | 15 | 30.04.2022 |
| 4.3 [Наличие и интуитивност на потребителски интерфейс (конзолен, графичен, уеб)](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Дизайн). | 10 | 31.05.2022 |
| 5. [Наличие и организация на автоматизирани тестове](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Тестване). | 15 | 15.06.2022 |
| 6. Организация на проекта в система за контрол на изходния код и употреба на добри практики (merge requests, code reviews, branching strategy) | 10 | 25.06.2022 |
| 1. [Въведение. Ниво на завършеност на проекта](file:///O:/Admin204.1/Проекти/2021-2022/Ученически%20практики%202/Ученическа%20документация/user_name_Software%20Development.docx#_Въведение) |  | 30.06.2022 |
| Документация на проекта (XML comments, wiki, etc.) | 5 | текущо |
| Презентация на проекта |  | 30.06.2022 |
| Общо | 100 | Финал на първа фаза |