**ПРЕДМЕТ: РАЗРАБОТКА НА СОФТУЕР**

**ДЕЙНОСТ: УЧЕНИЧЕСКИ ПРАКТИКИ 2**

***2022-2023***

**Тема: Конзолно приложение за графици на служители в предприятие**

**Курсов проект**

*Автори:*

*Теодора Николаева Компанска, XI клас*

Бургас

Съдържание

[1 Въведение 4](#_Toc136122745)

[2 Цели и обхват на софтуерното приложение 4](#_Toc136122746)

[2.1 Идея на проекта 4](#_Toc136122747)

[2.2 Опции на приложението 4](#_Toc136122748)

[2.3 Какъв проблем решава приложението? 5](#_Toc136122758)

[3 Анализ на решението 5](#_Toc136122759)

[3.1 Потребителски изисквания и работен процес 5](#_Toc136122760)

[3.2 Примерен потребителски интерфейс 6](#_Toc136122761)

[3.3 Диаграми на анализа 7](#_Toc136122764)

[3.4 Модел на съдържанието / данните 11](#_Toc136122769)

[4 Дизайн 11](#_Toc136122774)

[4.1 Реализация на структура на приложението (3-layer), Разделение на кода според предназначението му 11](#_Toc136122775)

[4.2 Организация и код на заявките към база от данни 12](#_Toc136122779)

[4.3 Наличие и интуитивност на потребителски интерфейс (конзолен, графичен, уеб) 12](#_Toc136122782)

[5 Тестване 13](#_Toc136122788)

[6 Заключение и възможно бъдещо развитие 13](#_Toc136122789)

[7 Използвани литературни източници и Уеб сайтове 14](#_Toc136122793)

[8 Критерии и показатели за оценяване 15](#_Toc136122794)

# Въведение

В много фирми и предприятия има тенденция на недостиг на началници, след което следва, че един началник трябва да управлява много работници по едно и също време, както и да оправя техните графици и отпуски. Това става твърде сложно, защото в повечето случаи заявяването на отпуск се случва вербално или чрез подаване на документи. Графиците, в повечето случаи, са изобразени на лист и всеки от работещите трябва поотделно да ги потърси или да се пращат поотделно на всеки.

Решението ми е да създам приложение, в което всичко това се случва дигитално и всеки работещ може да заяви отпуск по всяко време, дори и у дома си.

# Цели и обхват на софтуерното приложение

## Идея на проекта

Идеята на проекта е да се създаде уеб приложение, чрез което служителите и ръководителите на дадена фирма ще виждат своя график, ще могат да заявяват отпуск и съответно ръководителите да я одобряват.

## Опции на приложението

### **Опции на потребителите**

#### Опция за виждане на оставащи часове основен отпуск

* Виждане на историята по дни кога служител е взел отпуск

#### Опция за заявяване и виждане на отпуск

* Служителят има право да си заяви отпуск, като първоначалният статус е "в изчакване" и се очаква ръководителят да го одобри

#### Опция за виждане на графика на колеги и своя собствен (кой е в отпуск, коя смяна е на работа)

* Служителят има право да вижда графика на колеги САМО от своя отдел

### **Опции на ръководителите**

#### Опция за добавяне, редактиране, виждане на потребители и даване на администраторска роля

#### Опция за добавяне, редактиране, виждане на всички отдели и на всички потребители в даден отдел

#### Опция за добавяне на график на потребител и на виждане на график на даден потребител

#### Опция за одобряване и отхвърляне на отпуск и виждане на отпуска на даден потребител

## Какъв проблем решава приложението?

Приложението е с цел улеснение, главно откъм страна на служителите, за да могат да управляват своето свободно време. Ръководителите на съответната фирма ще бъдат улеснени от страна на одобряване на отпуск на служителите, тъй като ще става по-бързо и ще могат да го правят по всяко време.

# Анализ на решението

## Потребителски изисквания и работен процес

Потребителите на приложението са:

* Началниците, които нанасят графиците и одобряват отпуски на работниците, добавят, редактират и им задават администраторска роля, както и ги добавят към съответен отдел в предприятието
* Работниците, които виждат графика си и този на колегите си от същия отдел и могат да заявяват отпуск

Всеки работник може да:

* Вижда своя график и графика на колегите си от своя отдел
* Заявява и вижда отпуск
* Вижда остатък часове от основния отпуск

Всеки началник може да:

* Одобрява отпуск
* Задава график на работниците
* Добавя, вижда и редактира акаунти на работници в системата
* Добавя, вижда и редактира отдели на предприятието

Приложението ще изпраща заявки до своята база от данни, ще ги запазва и ще връща обработената информация в приложението.

## Примерен потребителски интерфейс

### **Страница за влизане в конзолно приложение**

A screenshot of a computer screen

Description automatically generated with medium confidence

### **Страница за влизане в мобилно приложение**



## Диаграми на анализа

### **Диаграма на случаите (Use Case Diagram)**

A picture containing text, diagram, sketch, drawing

Description automatically generated

Потребител (**User**) има достъп до опциите за отпуск и график, като може да заявява и вижда отпуската си и своя график, както и този на колегите от отдела си.

Администратор (**Admin**) има достъп до всички функции на приложението, като може да управлява потребителите, отделите, графиците, ролите и отпуските.

### **Модел на съдържанието / данните (Entity Relationship Diagram)**

A picture containing text, diagram, plan, parallel

Description automatically generated

Таблицата “Users” е главната таблица, към която другите са свързани. Един потребител може да има много графици и отпуски.

Таблицата „Schedules“ е таблицата за графиците, в която се съдържа информацията за датите и часовете на определения график за потребителя.

Таблицата “Vacations” е таблицата за отпуски, в която се съдържа информацията за датите на отпуската, както и описанието и за опреден отпуск на потребител.

Таблиците “Departments” и “Roles” са за отделите и ролите, към които потребителите на приложението принадлежат. Таблиците нямат външни ключове, но таблицата “Users” съдържа външни ключове към тях.

### **Диаграма на последователността и комуникацията (Sequence Diagram) – само за потребител, който не е администратор**

A picture containing text, document, parallel, diagram

Description automatically generated

Потребителят изпраща заявка за влизане в приложението, след като акаунтът му се потвърди, може да види опциите си за графици и отпуски. След което може да прати заявка за виждане на своя график или този на колегите си. Също така има опция за виждане или заявяване на отпуск, след което трябва да изчака началника (администратора) на приложението да потвърди или да отхвърли.

### **Диаграма на класовете (Class Diagram)**

A picture containing text, diagram, sketch, technical drawing

Description automatically generated

Класовете, които използвам са разделени на три части:

* Клас за данните от базата данни
* Клас за обработката на данните
* Клас за визуализиране на данните

## Модел на съдържанието / данните

### **Модел на конзолното приложение**

#### Входни данни:

* Потребителски вход: Потребителят въвежда информация от клавиатурата, като текстови стойности, числа и други.
* Входни параметри: Приложението може да получи входни параметри през командния ред при стартирането му.

#### Обработка на данни:

* Логика на приложението: В този модел, логиката на приложението включва обработка на входните данни, изчисления, манипулации с текстови стойности и списъци, валидации и други действия върху данните.
* Управление на потока: Конзолното приложение използва оператори за контрол на потока (if-else, цикли и др.) за вземане на решения, повторение на действия или изход от приложението.

#### Изходни данни:

* Конзолен изход: Резултатите от обработката на данните и логиката на приложението са изведени на конзолата, като текстови съобщения, резултати от изчисления или други информационни съобщения.
* Запис на данни: Приложението записва данните в база от данни

# Дизайн

## Реализация на структура на приложението (3-layer), Разделение на кода според предназначението му

### **Потребителски интерфейс (Presentation Layer):**

* Този слой се отнася до визуализацията и взаимодействието с потребителя.
* Включва всички компоненти, свързани с представянето на информацията на потребителя и получаването на входни данни от него.
* Използва библиотеки и методи за работа с конзолата, извеждане на текст, форматиране на данни и приемане на вход от клавиатурата.

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

* Този слой съдържа цялата бизнес логика и логиката на приложението.
* Включва обработка на входни данни, изчисления, валидации, манипулации със стрингове и листове, взаимодействие със съхранението на данни и други операции, свързани със специфичната функционалност на приложението.
* Използват се библиотеки и методи, които подпомагат обработката и манипулацията на данните, изпълнението на бизнес правила и други сходни операции.

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

* Този слой се грижи за взаимодействието със съхранението на данни
* Включва методи и библиотеки за достъп до данни, извличане, запис и промяна на информацията в съответните източници.
* Използва библиотеката Entity Framework

## Организация и код на заявките към база от данни

### **Извличане на обекти (Querying):**

* За да се извлекат данни от базата данни, се използват методите FirstOrDefault, SingleOrDefault, ToList и други върху DbSet (таблица) в контекста на базата данни.
* Тези методи позволяват формирането на заявки (queries), които да извлекат определени обекти, отговарящи на условия, като например филтриране по стойности на полета или присъствие на релационни връзки.
* Резултатът от заявката може да бъде единичен обект или списък от обекти, в зависимост от използвания метод.

### **Добавяне на обекти (Adding):**

* За добавянето на нови обекти в базата данни, се използва метода Add върху DbSet (таблица) в контекста на базата данни.
* Подготвят се новите обекти, като се задават стойности на техните свойства, и след това се добавят с помощта на Add метода.
* След добавянето на обектите се извиква метода SaveChanges на контекста на базата данни, за да се запишат промените в базата.

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

### **Регистрация и вход:**

* Началниците могат да влизат и да създават нови акаунти
* Всички потребители могат да влизат в системата, въвеждайки своите потребителско име и парола.

### **Меню за навигация:**

* Предоставя възможност за избор на различни опции и функционалности в приложението.
* Това включва визуализация на налични команди и менюта, които потребителят може да избира и използва.

### **Визуализация на данни:**

* Показване на информация и данни, свързани с потребителски профили, налични ресурси, резултати от заявки и други.
* Тези се извличат от базата данни и се представят на потребителя в удобен и разбираем формат.

### **Въвеждане и редактиране на данни:**

* Възможност за въвеждане на нови данни или редактиране на съществуващи данни.
* Това бива посредством въвеждане на текстови стойности, избиране от списъци, въвеждане на числови стойности и други.

### **Обработка на заявки и операции:**

* Изпълнение на заявки и операции, които се изискват от потребителя.
* Това включва изпращане на заявки към базата данни, обработка на данни, изчисления и други операции.

# Тестване

* Функционално тестване: Проверка на всяка функционалност на приложението, за да се уверим, че работи правилно и отговаря на очакванията. Това включва входни данни, изходни резултати и проверка на очаквания резултат.
* Интеграционно тестване: Тестване на взаимодействието между различните компоненти и модули на приложението. Целта е да се провери, че те работят правилно заедно и взаимодействат по предвиден начин.
* Производителностно тестване: Измерване на производителността и ефективността на приложението при различни натоварвания и обеми на данни. Целта е да се установи дали приложението работи с оптимална скорост и да се идентифицират евентуални проблеми с производителността.
* Сигурностно тестване: Проверка на сигурността на приложението, за да се открият и поправят възможни уязвимости и проблеми със защитата на данните. Това включва тестване на уязвимости, автентикация, авторизация и други аспекти на сигурността.

# Заключение и възможно бъдещо развитие

Проектът успешно реализира идеята си за улесняване на служителите и ръководителите в отношение на работното време. Приложението предоставя възможности за следене на работното време на служителите, заявяване и одобряване на отпуски, както и виждане на графика на колегите.

### **Предимства на използването на конзолно приложение с C#:**

* Скорост и ефективност: Конзолните приложения се изпълняват локално и обикновено имат бърз отговор, което ги прави подходящи за бързи и прости операции.
* Простота на използване: Конзолните приложения са лесни за разбиране и използване, особено за потребители, запознати с команден ред и конзолна интеракция.
* Гъвкавост и контрол: Конзолните приложения предоставят голям контрол върху входните и изходните данни, което позволява разработчиците да настройват и персонализират функционалността според нуждите на потребителите.

### **Ограничения на използването на конзолно приложение с C#:**

* Ограничени визуални възможности: Конзолните приложения са ограничени във визуалните елементи и интеракцията с потребителя. Те не предоставят графичен интерфейс и изискват потребителите да въвеждат команди чрез команден ред.
* Ограничена мултимедийна поддръжка: Конзолните приложения не са подходящи за работа с графика, аудио и видео, което ги прави несъвместими със сложни мултимедийни функции.

### **Бъдещо развитие**

* Интегриране на графичен потребителски интерфейс: Добавянето на GUI би подобрило използваемостта на приложението и предоставило по-интуитивен начин за взаимодействие с потребителите.
* Разширение на функционалността: Може да се разгледат допълнителни функционалности като автоматично изчисляване на работните часове, генериране на отчети, визуализация на статистики и други, които биха били полезни за потребителите.
* Мобилно приложение: Разработката на мобилно приложение би позволила на потребителите да достъпват и управляват работното време през своите мобилни устройства, което е удобно и популярно в днешно време.

# Използвани литературни източници и Уеб сайтове

* (Въведение в .Net)
* (Microsoft)

# 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 |