2022 - 2023

МАРИЯ Ж. ИЛЧЕВА

КУРСОВ ПРОЕКТ

БУРГАС

Система за управление на салон за красота

Документация

Съдържание

[1. Въведение 2](#_Toc130660811)

[2. Цели и обхват на софтуерното приложение 2](#_Toc130660812)

[3. Анализ на решението 2](#_Toc130660813)

[a. Потребителски изисквания и работен процес 2](#_Toc130660814)

[b. Примерен потребителски интерфейс 2](#_Toc130660815)

[c. Диаграми на анализа 2](#_Toc130660816)

[d. Модел на съдържанието/данните 3](#_Toc130660817)

[4. Дизайн 3](#_Toc130660818)

[a. Реализация на структура на приложението (3-layer), Разделение на кода според предназначението му 3](#_Toc130660819)

[b. Организация на код и заявките към базата от данни 3](#_Toc130660820)

[c. Наличие и интуитивност потребителски интерфейс (конзолен, графичен, уеб) 3](#_Toc130660821)

[5. Тестване 3](#_Toc130660822)

[6. Заключение и възможно бъдещо развитие 3](#_Toc130660823)

[7. Използвани литературни и Уеб сайтове 4](#_Toc130660824)

[8. Приложения 4](#_Toc130660825)

# Въведение

Настоящият документ представлява документация по проекта на Мария Живкова Илчева от XI В клас – „Ученически практики 2“. В него е представена всичката информация по проекта и неговата разработка.

# Цели и обхват на софтуерното приложение

Приложението представлява система за управление на салони за красота. В него ще могат да се записват на часове в различни регистрирани в системата козметични салони, с опцията да се прегледат всички възможни часове, с максимално време за размисъл и избор на специалист, който да извърши избраната от клиента процедура.

# Анализ на решението

## **Потребителски изисквания и работен процес**

За разглеждането на регистрираните салони, сайта и свободните часове, потребителят не се нуждае от нищо. За да си запази час или да изтрие вече запазен такъв, потребителят се нуждае от акаунт.

## Text Description automatically generated with medium confidence**Примерен потребителски интерфейс**

## **Диаграми на анализа**

* **Activity diagram**

Диаграмата на дейност визуално представя поредица от действия или поток от контрол в система, подобна на блок-схема или диаграма на потока от данни. Тези диаграми могат също така да описват стъпките в диаграма на случай на употреба. Моделираните дейности могат да бъдат последователни и едновременни. И в двата случая диаграмата на активността ще има начало (начално състояние) и край (крайно състояние).

* **Class diagram**

Илюстрация на връзките и зависимостите на изходния код между класовете в Unified Modeling Language (UML). В този контекст класът дефинира методите и променливите в обект, който е конкретен обект в програма или единица код, представляващ този обект. Диаграмите на класове са полезни във всички форми на обектно-ориентирано програмиране (ООП). Концепцията е на няколко години, но е усъвършенствана с развитието на парадигмите за ООП моделиране.

* **ER diagram**

Диаграмата на връзката между обекти (ER) е вид блок-схема, която илюстрира как „субекти“, като хора, обекти или концепции, се свързват помежду си в рамките на една система. ER диаграмите 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. Известни също като ERD или ER модели, те използват дефиниран набор от символи като правоъгълници, диаманти, овали и свързващи линии, за да изобразят взаимосвързаността на обекти, връзки и техните атрибути. Те отразяват граматическата структура, с обекти като съществителни и връзки като глаголи.

* **Use case diagram**

Диаграмите на случаи на използване описват функциите на високо ниво и обхвата на системата. Тези диаграми също така идентифицират взаимодействията между системата и нейните участници. Случаите на употреба и актьорите в диаграмите на случаите на употреба описват какво прави системата и как участниците я използват, но не и как системата работи вътрешно.

## **Модел на съдържанието/данните**

Данните от приложението ще се съхраняват в база от данни, направена по модела от ER диаграмата. В тази база не само ще се записват данни, но ще се и извличат такива. Съдържанието им ще варира от имена на служители на даден салон, тяхна лична информация, цени на услуги, използвани материали, до свободни часове при различните специалисти.

# Дизайн

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

В приложението можем да видим разделение на кода на приложението в съответствие с 3-слойната архитектура. В различните файлове и класове можем да забележим следните слоеве:

DAL (Data Access Layer): Класовете и функциите в папката "bsm.dal" се отнасят до достъпа до данни и комуникацията с базата данни. Те са отговорни за извличане, добавяне, редактиране и изтриване на данни от базата данни. Примери на това включват класовете "UserService" и "ServiceGroupService", които извършват операции свързани с потребителите и групите на услуги съответно.

BLL (Business Logic Layer): Класовете и функциите в папката "bsm.bll" съдържат бизнес логиката на приложението. Те са отговорни за обработката на бизнес правила и логиката на приложението, използвайки функционалността предоставена от DAL. Примери на това включват класовете "UserService" и "ServiceGroupService", които съдържат функции за валидация, обработка на данни и извикване на методи от DAL слоя.

Console (Presentation Layer): Класовете и функциите в папката "bsm.console" са отговорни за конзолния потребителски интерфейс на приложението. Те съдържат функции за показване на информация и обработка на потребителски вход. Примери на това включват класовете "StartMenu", "MainMenu", "ServiceGroupEditMenu" и други, които се отнасят до показване на менюта, обработка на избор на потребителя и извикване на съответните функционалности от BLL слоя.

## **Организация на код и заявките към базата от данни**

Кодът е организиран по начин, който следва добри практики за организацията на кода и заявките към базата данни. Има ясно разделение между слоевете на архитектурата и отговорностите им. DAL слоят е отговорен за взаимодействието с базата данни, като използва подходящи ORM (Object-Relational Mapping) инструменти или заявки към базата данни, за да извършва операции с данни. BLL слоят се грижи за бизнес логиката на приложението, като извиква подходящите функции от DAL слоя за достъп до данните.

c. Наличие и интуитивност на потребителски интерфейс (конзолен, графичен, уеб):

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

В предоставения код се използва конзолен потребителски интерфейс (CLI). Менюта и опциите се показват на конзолата, а потребителят може да избира опции чрез въвеждане на съответни клавиши. Въпреки че конзолният интерфейс не предоставя визуални елементи и интерактивността на графичния или уеб интерфейс, той е интуитивен за използване и предоставя прост начин за взаимодействие с приложението.

Общо взето, предоставеният код демонстрира добра организация на кода, разделение на слоевете и прилагане на 3-слойната архитектура. Имайки конзолен потребителски интерфейс, приложението предоставя основни функционалности и възможности за работа с потребителите и групите на услуги.

# Тестване

Тестването е важна част от разработката на софтуер, което ни помага да проверим функционалността, надеждността и изпълнението на нашия проект. Ето основните аспекти по тестването на проекта

* Видове тестове:

- Модулни тестове: Тези тестове се фокусират върху отделни модули или компоненти от приложението и се изпълняват на ниво код. Те проверяват дали всеки модул работи коректно и отделно от останалите части на системата.

- Интеграционни тестове: Тези тестове се изпълняват за да се провери взаимодействието и комуникацията между различните модули и компоненти на приложението.

- Функционални тестове: Тези тестове се използват за проверка на цялостната функционалност на приложението, като се проверява дали системата изпълнява очакваните функции и отговаря на изискванията.

- Приемателни тестове: Тези тестове се извършват от крайните потребители или клиенти на приложението с цел проверка на функционалността и приемане на системата.

* Създаване на тестови случаи:

- Идентифициране на основните функционалности и сценарии, които трябва да бъдат тествани.

- Създаване на тестови случаи, които покриват различни комбинации от входни данни, гранични случаи и негативни сценарии.

- Дефиниране на очакваните резултати и поведение за всеки тестов случай.

* Изпълнение на тестовите случаи:

- Изпълнение на тестовите случаи, следвайки зададените стъпки и входни данни.

- Запис на фактическите резултати и поведение на приложението по време на тестването.

* Анализ на резултатите:

- Сравнение на фактическите резултати с очакваните резултати.

- Идентифициране и отстраняване на грешки и проблеми.

- Оценка на преминаването на тестовете и качеството на приложението.

Тестването е непрекъснат итеративен процес, който може да се изпълнява в различни фази на разработката на проекта. Той помага да се гарантира, че приложението отговаря на изискванията и работи правилно в различни сценарии и условия.

# Заключение и възможно бъдещо развитие

Приложението ще помогне на по-ангажираните и на по-свенливите хора да си запазват часове за различни процедури. Отваря се неограничено време за размисъл, без никакъв стрес. Самата идея може да бъде имплементирана в много други свери за в бъдеще като например за поправка на коли, майсторски услуги като боядисване, шлифоване и други. Най-доброто, което можем да постигнем е да усъвършенстваме и да осъществим идеята и тя наистина да бъде използвана от целевата аудитория и от още много други.

# Използвани литературни и Уеб сайтове

* Microsoft Documentation: https://docs.microsoft.com/

Официалната документация на Microsoft предлага подробни ръководства, уроци и примери за различни аспекти на разработката на софтуер с помощта на Microsoft технологии и инструменти.

* Stack Overflow: https://stackoverflow.com/

Stack Overflow е популярен форум, където програмисти от цял ​​свят публикуват въпроси и отговори на различни теми. Можете да търсите въпроси и отговори, свързани със сферите на вашите интереси и да се възползвате от опита и знанията на други програмисти.

* GitHub: https://github.com/

GitHub е популярна платформа за управление на изходен код и сътрудничество. Можете да търсите проекти, подобни на вашия, и да разглеждате изходния код, проблемите, документацията и други ресурси, които могат да ви дадат идеи и насоки за вашата работа.

* Software Testing Help: https://www.softwaretestinghelp.com/

Software Testing Help е онлайн платформа, където можете да намерите статии, уроци, насоки и ръководства за различни аспекти на тестването на софтуер. Те предлагат информация за различни методи, инструменти и подходи към тестването.

* Testing Excellence: https://www.testingexcellence.com/

Testing Excellence е уебсайт, който се фокусира върху различни аспекти на тестването на софтуер, включително методи, стратегии, инструменти и добри практики. Те предоставят статии, ръководства и туториали, които могат да бъдат полезни за разбиране на тестването и неговото приложение във вашия проект.

# Приложения

Приложението, което обсъждаме, е базирано на конзолен интерфейс и се фокусира върху управлението на групи за услуги. Ето няколко приложения на това приложение в ежедневието:

Управление на бизнес услуги: Приложението може да се използва от собствениците на бизнеси, които предлагат различни услуги (например салони за красота, спа центрове, фитнес студиа и др.) За тях приложението може да бъде полезно за управление на групите за услуги, включително добавяне, редактиране и изтриване на групи.

Резервации за услуги: Приложението може да се използва от клиенти, които искат да направят резервации за определени услуги. Клиентите могат да използват приложението за преглед на наличните групи за услуги и извършване на резервации за тях.

Управление на работници: Ако бизнесът има служители или работници, приложението може да се използва за управление на техните графици или работни дни. Ръководителите могат да използват приложението за преглед на заетостта на работниците и за даване на достъп до групите за услуги, които те трябва да обслужват.

Организиране на събития: Приложението може да се използва за организиране на събития, където се предлагат различни услуги. Например, приложението може да бъде полезно за организатори на конференции, фестивали или работилници, които искат да създадат групи за различни видове услуги, като лекции, демонстрации или консултации.

Планиране на лично време: Потребителите могат да използват приложението за планиране на своето лично време и управление на групи от дейности. Например, могат да създадат групи за различни хобита, спортни занимания или домашни задачи и да организират своето време въз основа на тези групи.