МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ВЫСОКИХ ТЕХНОЛОГИЙ И ПЬЕЗОТЕХНИКИ

Кафедра прикладной информатики и инноватики

КУРСОВАЯ РАБОТА

по дисциплине

«Архитектура информационных систем»

на тему

«Сервис для мастерской штанов на заказ»

Выполнил

студент 4 курса 3 группы

Никитин Владислав Алексеевич

Принял

профессор, д.т.н.

Жмайлов Борис Борисович

Ростов-на-Дону

2023

1. Техническое задание

Предметная область курсовой работы связана с пользователями и информационными ресурсами, связанными с компонентами создания штанов. Проект должен представлять из себя библиотеку в формате jar архива.

Необходимо создать сервис для мастерской штанов на заказ. Проект должен представлять собой Java приложение, предоставляющее API (application programming interface).

Предусмотреть в системе базовые операции: создание, редактирование, удаление и получение основных сущностей, таких как штаны и их свойства и материалы(ткань, цвета, нашивки, размер).

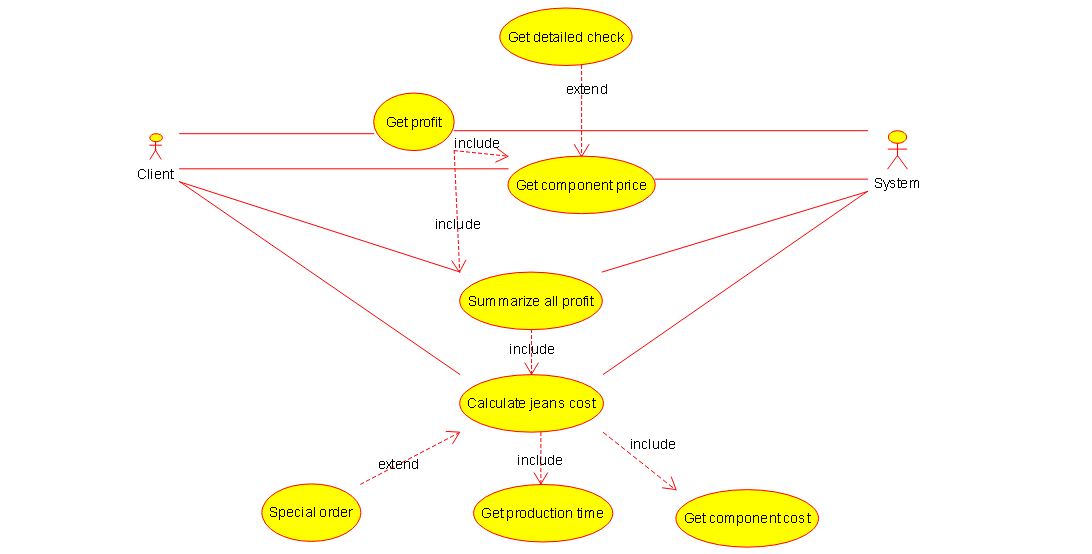
1. UML моделирование
   1. Диаграмма вариантов использования

Схема 1. Диаграмма вариантов использования

Таблица 1. Детализация диаграммы вариантов использования.

|  |  |
| --- | --- |
| Get component price | ***Включенный метод.*** *Входные**данные передаются в сигнатуре родительского метода getProfit().*  Цель: получить цену всех текущих компонентов на складе, если у пользователя есть доступ.  **Входные данные:**   * User user * boolean typeOfPrice   **Выходные данные:**   * Объект DataOfResult <Integer>   *Расширяем методом* **getDetailedCheck**.  Условие: Метод вызывается, если поле родительского метода typeOfPrice равно «DETAIL».  Параметры: *Параметры передаются в сигнатуре метода getСomponentPrice.* |
| Get detailed check | Цель: получить детальную информацию о всех текущих товарах на складе  **Входные данные:**  **Выходные данные:**   * Объект DataOfResult <List <Сomponent>> |
| Get profit | Цель: получить разницу между затраченными на товары деньгами и полученной прибылью, если у пользователя есть доступ.  **Входные данные:**   * User user * boolean typeOfPrice   **Выходные данные:**   * DataOfResult <Integer> |
| Summarise all profit | ***Включенный метод.*** *Входные**данные передаются в сигнатуре родительского метода getProfit().*  Цель: высчитать прибыль на основе стоимости джинс и затрат времени на их производство, если у пользователя есть доступ.  **Входные данные:**   * User user   **Выходные данные:**   * DataOfResult <Integer> |
| Calculate jeans cost | ***Включенный метод.*** *Входные**данные передаются в сигнатуре родительского метода summariseAllProfit().*  Цель: Метод позволяет рассчитать стоимость заказанных пользователем джинс  **Входные данные:**   * Jeans jeans   **Выходные данные:**   * DataOfResult <Integer>   *Расширяем методом* **specialOrder**.  Условие: Метод вызывается, если классом джинс является CustomOfferJeans.  Параметры: *Параметры передаются в сигнатуре метода Calculatejeanscost.* |
| Special order | Цель: метод возвращает уже рассчитанную стоимость джинс, без потребности рассчитывать заказ с самого начала  **Входные данные:**   * Jeans jeans   **Выходные данные:**   * DataOfResult <Integer> |
| Get production time | ***Включенный метод.*** *Входные**данные передаются в сигнатуре родительского метода Calculatejeanscost().*  Цель: метод возвращает время на создание джинс.  **Входные данные:**  **Выходные данные:**   * DataOfResult <Integer> |
| Get component cost | ***Включенный метод.*** *Входные**данные передаются в сигнатуре родительского метода Calculatejeanscost().*  Цель: Метод возвращает стоимость всех компонентов для джинс  **Входные данные:**   * CustomOfferJeans Jeans   **Выходные данные:**   * DataOfResult <Integer> |

* 1. Диаграмма классов

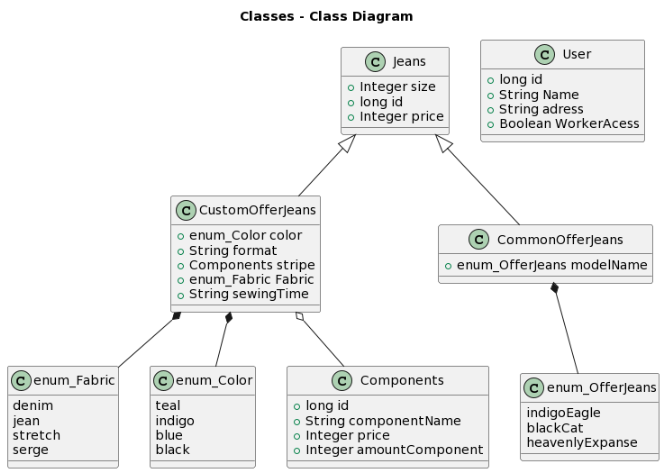
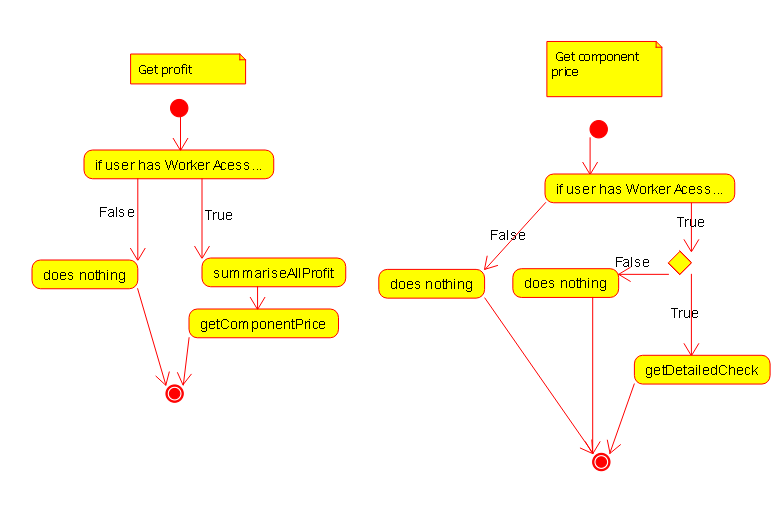


Схема 2. Диаграмма классов

* 1. Диаграммы активностей

Схема 3.1 Диаграмма активностей

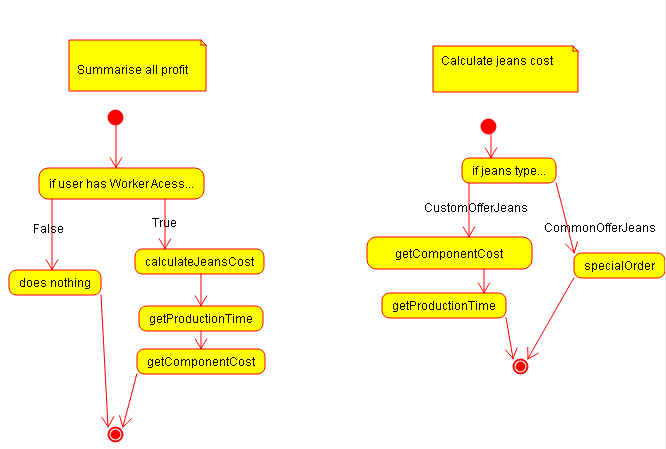


Схема 3.2 Диаграмма активностей

* 1. Диаграмма компонентов

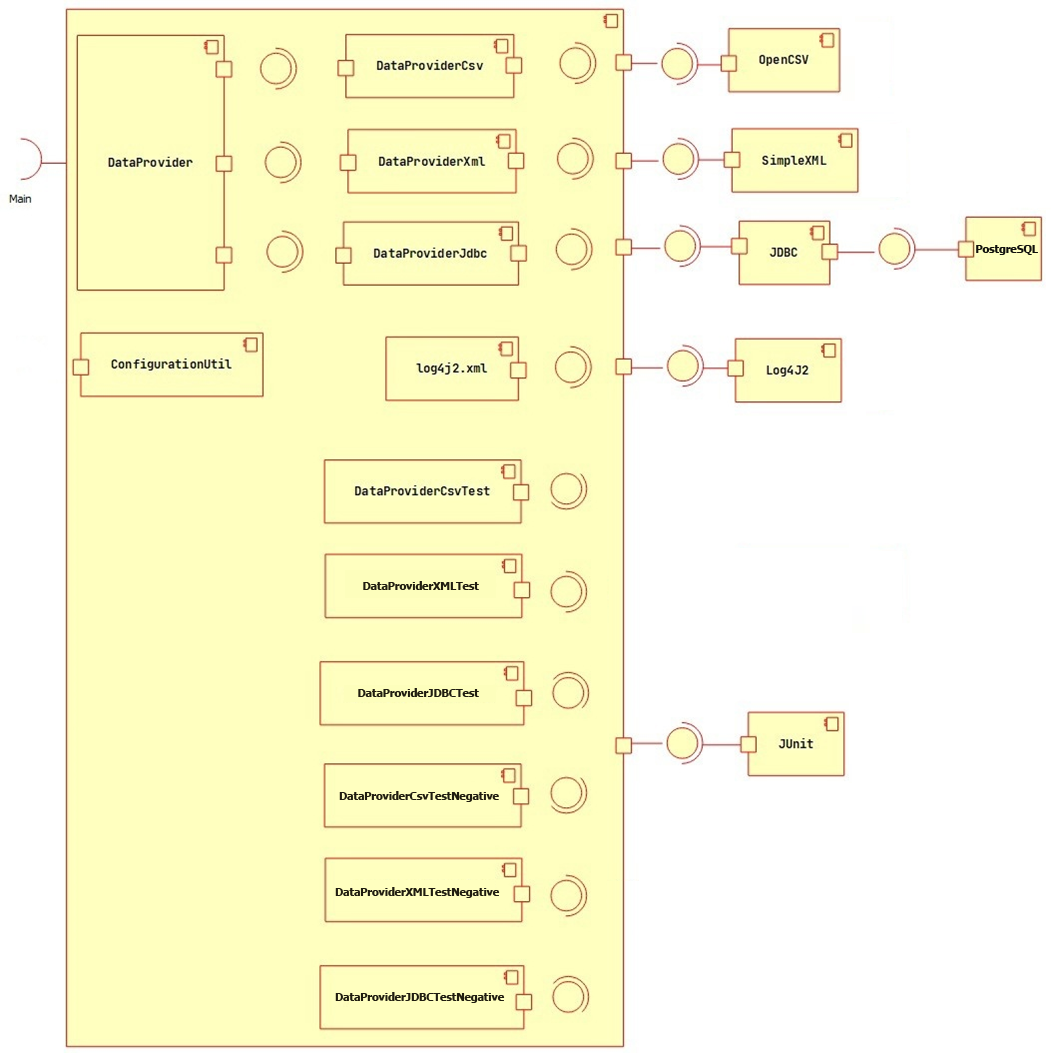


Схема 4. Диаграмма компонентов

Таблица 2. Сценарии и результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Тестовый метод** | **Тип сценария** | **Тестируемый метод** | **Что контролируется?** |
| GetDetailedCheckTest | позитивный | DataBasePantsworkshop.getDetailedCheck | assertEquals(TestData.components, DataBasePantsworkshop.getDetailedCheck().getObject()) |
| GetDetailedCheckTestNegative | негативный | DataBasePantsworkshop.getDetailedCheck | assertEquals("select empty Error", DataBasePantsworkshop.getDetailedCheck().getMessage()) |
| SpecialOrderTest | позитивный | DataBasePantsworkshop.specialOrder | assertEquals(TestData.commonOfferJeans1.getPrice(), DataBasePantsworkshop.specialOrder(TestData.commonOfferJeans1).getObject()) |
| SpecialOrderTestNegative | негативный | DataBasePantsworkshop.specialOrder | assertEquals("jeans model incorrect", DataBasePantsworkshop.specialOrder(new CommonOfferJeans()).getMessage()) |
| GetProductionTimeTest | позитивный | DataBasePantsworkshop. getProductionTime | assertEquals(TestData.customOfferJeans1.getSewingTime(), DataBasePantsworkshop.getProductionTime(TestData.customOfferJeans1).getObject()) |
| GetProductionTimeTestNegative | негативный | DataBasePantsworkshop.getProductionTime | assertEquals("jeans model incorrect", DataBasePantsworkshop.getProductionTime(new CustomOfferJeans()).getMessage()) |
| GetComponentPriceTest | позитивный | DataBasePantsworkshop.getComponentPrice | assertEquals(262, DataBasePantsworkshop.getComponentPrice(TestData.User1, false).getObject()) |
| GetComponentPriceTestNegative | негативный | DataBasePantsworkshop.getComponentPrice | assertEquals("worker access required", DataBasePantsworkshop.getComponentPrice(TestData.User2, false).getMessage()) |
| GetProfitTest | позитивный | DataBasePantsworkshop.getProfit | assertEquals(3525, DataBasePantsworkshop.getProfit(TestData.User1, false).getObject()) |
| GetProfitTestNegative | негативный | DataBasePantsworkshop.getProfit | assertEquals("worker access required", DataBasePantsworkshop.getProfit(TestData.User2, false).getMessage()) |
| CalculateJeansCostTest | позитивный | DataBasePantsworkshop.calculateJeansCost | assertEquals(4888, DataBasePantsworkshop.calculateJeansCost(TestData.customOfferJeans1).getObject()) |
| CalculateJeansCostTestNegative | негативный | DataBasePantsworkshop.calculateJeansCost | assertEquals("jeans model incorrect", DataBasePantsworkshop.calculateJeansCost(new CustomOfferJeans()).getMessage()) |
| GetComponentCostTest | позитивный | DataBasePantsworkshop.getComponentCost | assertEquals(44, DataBasePantsworkshop.getComponentCost(TestData.customOfferJeans1).getObject()) |
| GetComponentCostTestNegative | негативный | DataBasePantsworkshop.getComponentCost | assertEquals("jeans model incorrect", DataBasePantsworkshop.getComponentCost(new CustomOfferJeans()).getMessage()) |
| SummariseAllProfitTest | позитивный | DataBasePantsworkshop.summariseAllProfit | assertEquals(3787, DataBasePantsworkshop.summariseAllProfit(TestData.User1).getObject()) |
| SummariseAllProfitTestNegative | негативный | DataBasePantsworkshop.summariseAllProfit | assertEquals("worker access required", DataBasePantsworkshop.summariseAllProfit(TestData.User2).getMessage() |

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лабораторный практикум по дисциплине "Архитектура информационных систем". проф. Жмайлов Б.Б. ЮФУ, ИВТиП, каф. ИИТ. URL: <https://vk.com/doc407893182_619361878?hash=7ff6d767c480a7fec4&dl=6e2cdd08454a12355a>
2. Официальная документация SimpleXML с примерами сериализации <URL:http://simple.sourceforge.net/download/stream/doc/examples/examples.php>
3. UML диаграмма классов. URL: <https://russianblogs.com/article/36871440093/>
4. Разбор XML при помощи Simple Framework URL: <https://habr.com/ru/post/116830/>
5. MongoDB. URL: <https://javarush.ru/groups/posts/2776-videouikend-29-podkljuchaem-java-proekt-k-mongodb-uznaem-kak-statjh-programmistom-kogda-tebe-uz>
6. Библиотека OpenCsv. URL: <https://techcave.ru/posts/99-rabotaem-s-failami-csv-v-java-s-ispolzovaniem-biblioteki-opencsv.html>
7. Разбор XML при помощи Simple Framework URL: <https://habr.com/ru/post/116830/>
8. Работа с MongoDB. URL: <https://metanit.com/nosql/mongodb/2.3.php>
9. Как считывать и парсить CSV файлы. URL: [https://mkyong.com/java/how-to-read-and-parse-csv-file-in-java](https://mkyong.com/java/how-to-read-and-parse-csv-file-in-java/)
10. Как работать со средствами обобщения URL: <https://javarush.com/groups/posts/2004-teorija-dzhenerikov-v-java-ili-gde-na-praktike-stavitjh-skobki>