

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники Направление подготовки
09.03.04 Программная инженерия Дисциплина «Информационные системы»

ОТЧЕТ

По лабораторной работе 3

Вариант 20004

Преподаватель:

Тюрин Иван Николаевич

Выполнил:

Ильин Никита

Группа:

P3310

Санкт-Петербург
2025 г.

Задание

Доработать ИС из ЛР2 следующим образом:

Осуществить подключение к базе данных через Connection Pool:
использовать Druid

в отчете к лабораторной работе описать параметры конфигурации, которые использовались для настройки пула соединений

Добавить L2 JPA Cache:

в качестве реализации использовать Ehcache

реализовать в приложении возможность включения/отключения логирования статистики об использовании L2 JPA Cache (cache hits, cache misses) Если ваша лабораторная работа основана на Spring - для осуществления данной возможности использовать AOP; если приложение на Jakarta EE - использовать CDI Interceptors

в отчете к лабораторной работе описать параметры конфигурации, которые вы использовали для настройки L2-кэша и как они влияют на уровень хранения

Реализовать сохранение загруженных на сервер файлов, используемых для импорта данных, в файловом хранилище MinIO (можно взять любое другое S3-совместимое хранилище). Поднять и настроить MinIO требуется самостоятельно. Загруженные файлы должны быть доступны для скачивания из таблицы с логом импорта.

Сохранение загруженных файлов в файловом хранилище должно быть реализовано транзакционно по отношению к операциям, реализующим непосредственную вставку объектов в БД при импорте.

Для реализации распределенной транзакции из пункта 2 разрешается использовать любые инструменты. Рекомендуется решать задачу при помощи собственной реализации двухфазного коммита на уровне бизнес-логики приложения.

Необходимо на защите быть готовым продемонстрировать корректность реализованной распределенной транзакции в следующих условиях:

отказ файлового хранилища (БД продолжает работать)

отказ БД (файловое хранилище продолжает работать)

ошибка в бизнес-логике сервера (работают и БД, и файловое хранилище, однако в коде сервера вылетает RuntimeException между запросами в разные источники данных)

Необходимо:

на защите быть готовым продемонстрировать корректность работы распределенной транзакции в условиях параллельных запросов от нескольких пользователей (реализованный в ЛР 2 сценарий для Apache JMeter, тестирующий функцию импорта, должен продолжать корректно обрабатывать)

в отчете привести примеры ситуаций, которые могут возникать при осуществлении параллельных запросов от нескольких пользователей в вашем приложении, и описать, как ваша реализация обрабатывает такие ситуации

После окончания демонстрации лабораторной работы ваше приложение и все запущенные элементы окружения, необходимые для работы приложения, должны быть остановлены.

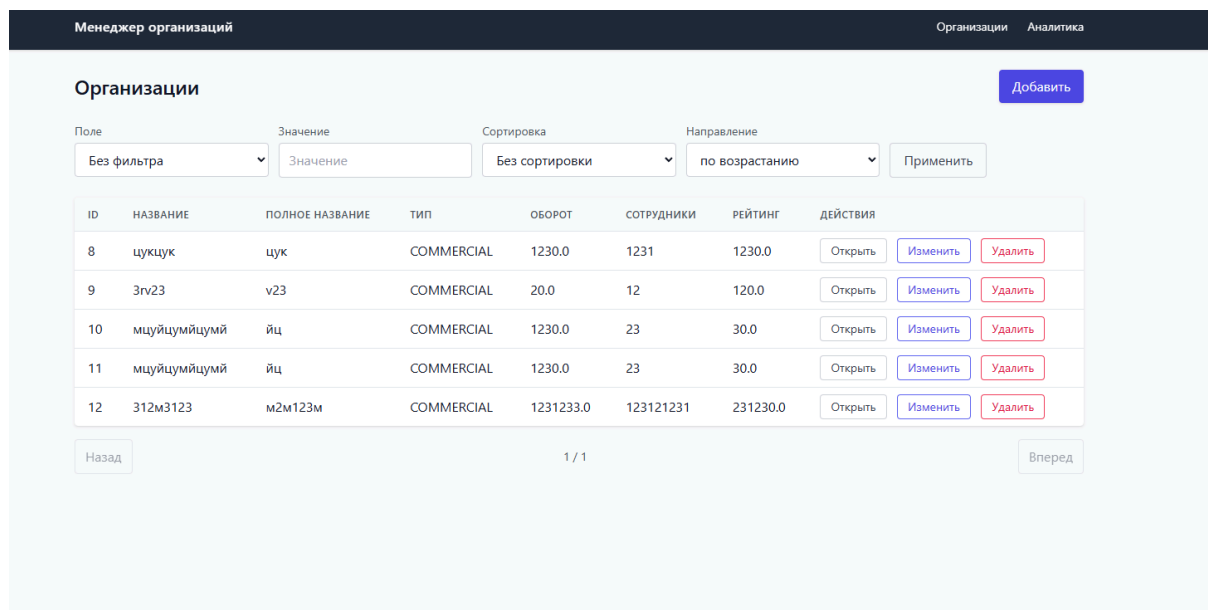


Рисунок 1: Пользовательский интерфейс разработанного приложения

Исходный код

https://github.com/MrTheFall/study_is

UML-диаграммы

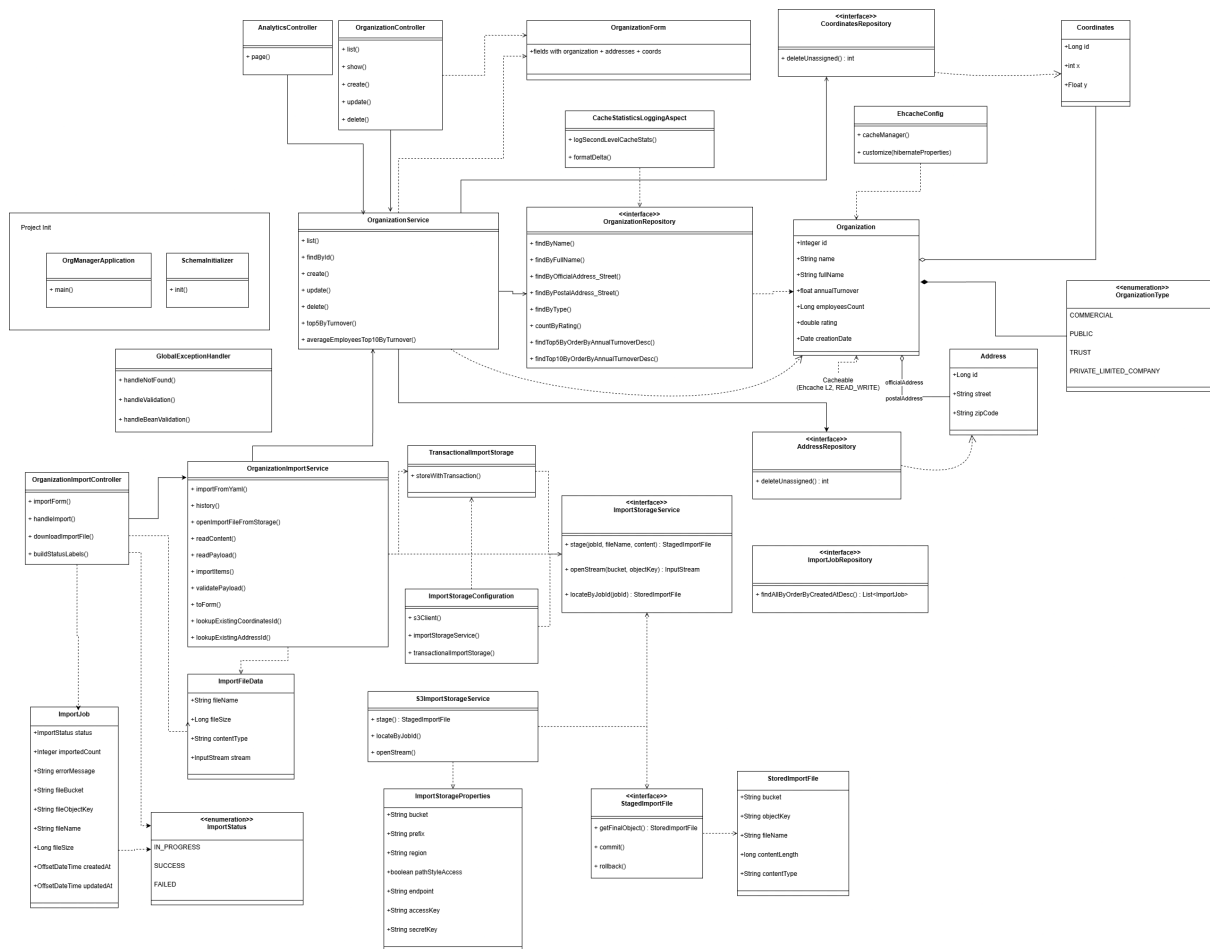


Рисунок 2: UML-диаграмма классов

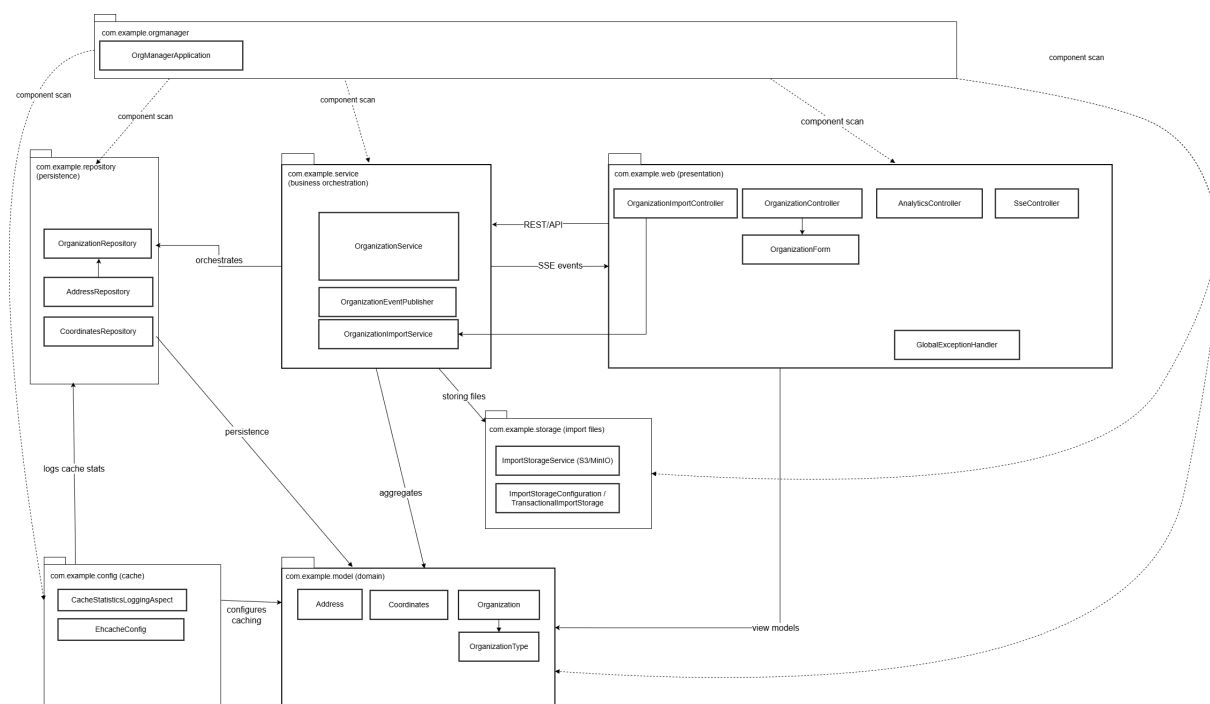


Рисунок 3: UML-диаграмма пакетов и их зависимостей

Выводы по работе

Проект получил поддержку массового импорта, с отказоустойчивым хранением импортируемых файлов и кешированием горячих данных.

На бэкенде организовано транзакционное сохранение импортируемых файлов в S3/MinIO с откатом при любой ошибке, а сами данные импортируются с учётом повторного использования координат и адресов. Подключён L2-кэш Hibernate на Ehcache для ускорения повторных запросов к организациям и добавлено аспектное логирование статистики кэша.

У интерфейса появился журнал импортов с возможностью скачать исходный файл напрямую из S3. В совокупности решение демонстрирует применение Spring MVC, JPA/Hibernate, S3-совместимых хранилищ и практик кеширования для повышения производительности и отказоустойчивости.