

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Информационные системы
Отчёт по лабораторной работе №3
Вариант 563187**

Выполнил: Зинченко Иван Николаевич
Группа: Р3312
ИСУ: 408657
Преподаватель: Райла Мартин

Санкт-Петербург
2026 год

Текст задания

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 Infinispan
 - 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 отрабатывать)
 - 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.

UML-диаграммы классов и пакетов разработанного приложения

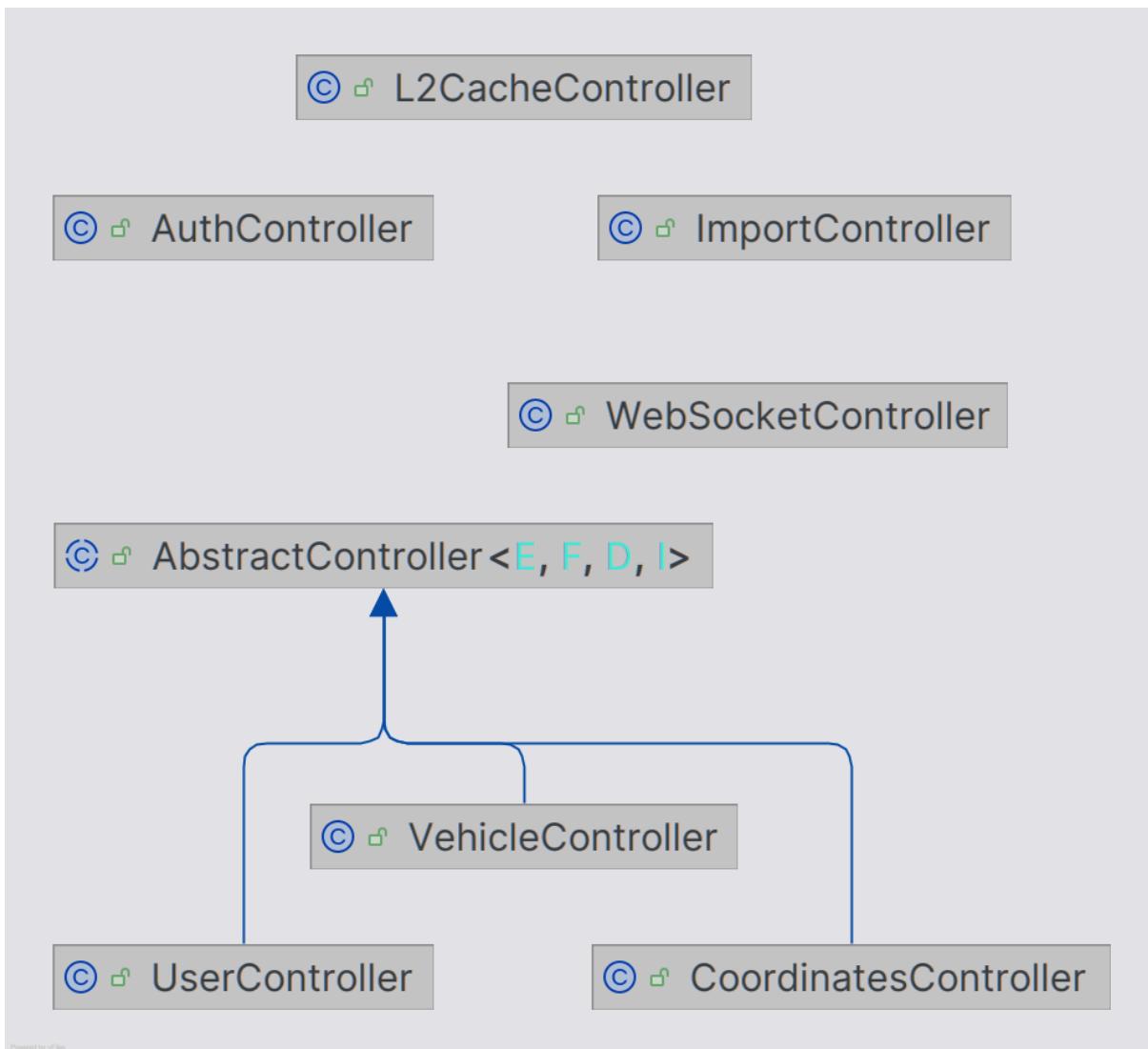


Рис. 1. контроллеры

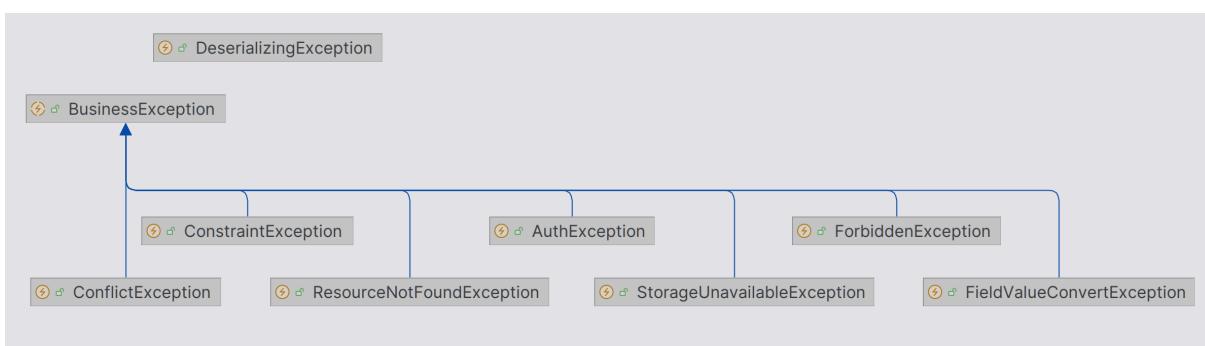


Рис. 2. исключения

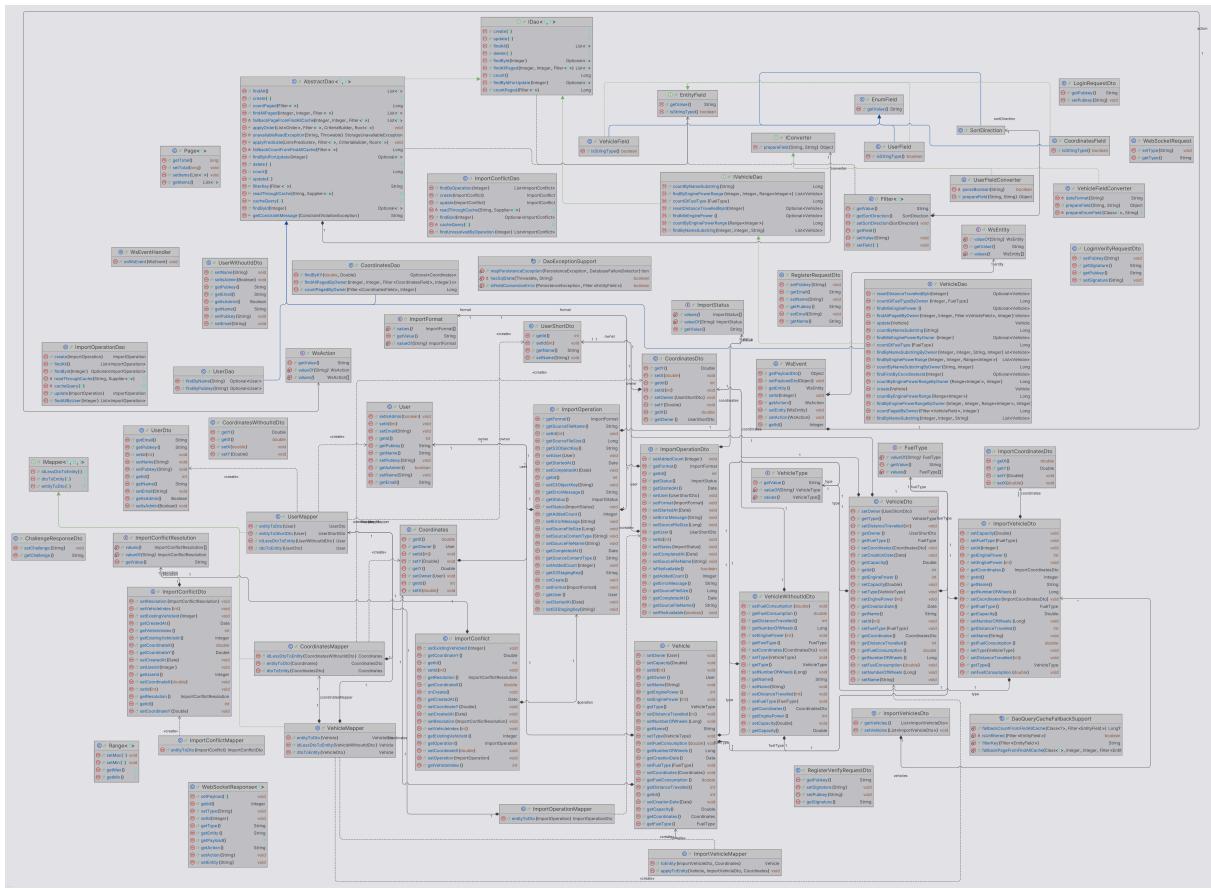


Рис. 3. модели



Рис. 4. серверные пакеты



Рис. 5. сервисы

Исходный код системы или ссылка на репозиторий с исходным кодом

<https://github.com/zinchenko291/is-lab-3>

Параметры конфигурации пула соединений

- name="PostgresDruidPool": имя пула в GlassFish.
- res-type="javax.sql.DataSource": тип ресурса для JPA/JNDI.
- datasource-classname="me.zinch.is.islab3.datasource.GlassfishDruidDataSource": класс DataSource.
- steady-pool-size="1": минимальное число соединений, которое GlassFish старается держать.
- max-pool-size="30": верхний лимит соединений.
- pool-resize-quantity="2": шаг, с которым пул расширяется/сужается.
- idle-timeout-in-seconds="300": через сколько секунд простоя соединение можно закрыть.
- connection-validation-method="auto-commit": метод валидации соединения на стороне GlassFish.
- is-connection-validation-required="true": валидация соединений включена.
- fail-all-connections="false": при сбое не инвалидировать весь пул разом.
- non-transactional-connections="true": разрешает использовать соединения вне JTA-транзакций.
- allow-non-component-callers="true": разрешает доступ к ресурсу вне контейнерных компонентов.

параметры Druid/PostgreSQL:

- `url`: JDBC URL + параметры драйвера.
- `user`: пользователь БД.
- `password`: пароль/секрет для БД.
- `driverClassName="org.postgresql.Driver"`: JDBC-драйвер PostgreSQL.
- `initialSize="1"`: стартовое число созданных коннектов Druid.
- `minIdle="1"`: минимум idle-коннектов в Druid.
- `maxActive="30"`: максимум активных коннектов Druid.
- `maxWait="3000"`: ждать коннект из пула до 3 сек.
- `loginTimeout="2"`: таймаут установления нового JDBC-коннекта.
- `validationQuery="SELECT 1"`: SQL для проверки живости соединения.
- `validationQueryTimeout="2"`: таймаут validation query.
- `testWhileIdle="true"`: проверять idle-коннекты в фоне.
- `testOnBorrow="true"`: проверять коннект перед выдачей.
- `testOnReturn="false"`: не проверять при возврате в пул.
- `timeBetweenEvictionRunsMillis="10000"`: каждые 10 сек запуск idle проверки.
- `minEvictableIdleTimeMillis="300000"`: idle-коннект может быть удален после 5 минут.
- `keepAlive="true"`: поддерживать соединения живыми.
- `failFast="false"`: не падать сразу при стартовых проблемах подключения.
- `breakAfterAcquireFailure="false"`: после неудачи продолжать попытки создания коннектов.
- `connectionErrorRetryAttempts="2147483647"`: число retry.
- `timeBetweenConnectErrorMillis="5000"`: пауза 5 сек между retry на создание коннекта.

JDBC URL-параметры:

- `stringtype=unspecified`: строковые параметры без жесткого типа (полезно для некоторых PG-кейсов).
- `connectTimeout=3`: таймаут TCP connect к БД 3 сек.
- `socketTimeout=5`: таймаут операций чтения/записи 5 сек.
- `tcpKeepAlive=true`: включить TCP keepalive на сокете.

Параметры конфигурации L2 кеша

- `name="eclipselink.cache.shared.default" value="true"`: включение кеша в приложении.
- `name="eclipselink.cache.size.default" value="1000"`: количество объектов для хранения в кеше при старте.
- `name="eclipselink.session.customizer" value="me.zinch.is.islab3.server.cache.InfinispanSessionCustomizer"`: указание на программно управляемый экземпляр кеша.

Выходы по работе

В ходе проделанной работы я узнал о распределённых транзакциях, кешировании в JPA и работе с пулом соединений. Была доработана ЛР 2, в интерфейс добавлена возможность выгружать ранее загруженные файлы. Сервер стал отказоустойчивым, продолжая обслуживать пользователей при отключении БД или S3 хранилища.