

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

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

ОТЧЕТ

По лабораторной работе №2

Вариант 1000

Преподаватель:

Тюрин Иван Николаевич

Выполнил:

Алхимовици Арсений

Группа:

P3310

Санкт-Петербург
2025 г.

Задание

Доработать ИС из ЛР1 следующим образом:

Добавить в систему возможность массового добавления объектов при помощи импорта файла. Формат для импорта необходимо согласовать с преподавателем. Импортируемый файл должен загружаться на сервер через интерфейс разработанного веб-приложения. При реализации логики импорта объектов необходимо реализовать транзакцию таким образом, чтобы в случае возникновения ошибок при импорте, не был создан ни один объект.

При импорте должна быть реализована проверка пользовательского ввода в соответствии с ограничениями предметной области из ЛР1.

При наличии вложенных объектов в основной объект из ЛР1 необходимо задавать значения полей вложенных объектов в той же записи, что и основной объект.

Необходимо добавить в систему интерфейс для отображения истории импорта (обычный пользователь видит только операции импорта, запущенные им, администратор - все операции).

В истории должны отображаться id операции, статус ее завершения, пользователь, который ее запустил, число добавленных объектов в операции (только для успешно завершенных).

Согласовать с преподавателем и добавить в модель из первой лабораторной новые ограничения уникальности, проверяемые на программном уровне (эти новые ограничения должны быть реализованы в рамках бизнес-логики приложения и не должны быть отображены/реализованы в БД).

Реализовать сценарий с использованием Apache JMeter, имитирующий одновременную работу нескольких пользователей с ИС, и проверить корректность изоляции транзакций, используемых в ЛР. По итогам исследования поведения системы при ее одновременном использовании несколькими пользователями изменить уровень изоляции транзакций там, где это требуется. Обосновать изменения.

Реализованный сценарий должен покрывать создание, редактирование, удаление и импорт объектов.

Реализованный сценарий должен проверять корректность поведения системы при попытке нескольких пользователей обновить и\или удалить один и тот же объект (например, двух администраторов).

Реализованный сценарий должен проверять корректность соблюдения системой ограничений уникальности предметной области при одновременной попытке нескольких пользователей создать объект с одним и тем же уникальным значением.

Согласованные ограничения:

1. One car can be owned by a maximum of 4 people
2. The soundtrack cannot be repeated
3. Coordinates, names and machines cannot be the same
4. Coordinates cannot be the coordinates of the user's city
5. You can't create a person with mood=rage if the user's current time is 19:00-10:00

HumanBeing Management

Search by name, soundtrack, or car...

Search

Clear

Special Operations

Sum Impact Speed

Delete Heroes w/o Toothpick

Assign Lada to Heroes

Import Humans (JSON)

Create Human

Name

X

Y (≤818)

Not Hero

No toothpick

Select car (required)

No mood

Impact

HAMMER

Soundtrack

Create Human

Create Car

Car name (optional)

Not cool

Create Car

Humans

ID	Name	Coords	Hero	Toothpick	Mood	Car	Impact	Soundtrack	Weapon	Actions
33	Bob	11.0,21.0	No	Yes	CALM	BMW-2	1.2	track-002	HAMMER	<div>Edit</div> <div>Delete</div>
32	Alice	10.0,20.0	Yes	No	RAGE	BMW-1	0.5	track-001	AXE	<div>Edit</div> <div>Delete</div>
29	TestUser1	100.5,200.5	Yes	Yes	RAGE	BMW-1	150.0	MySoundtrack	AXE	<div>Edit</div> <div>Delete</div>
3	TestUserUpdated	132.0,133.0	Yes	Yes	RAGE	TestCar	132.0	123tesxt	HAMMER	<div>Edit</div> <div>Delete</div>

Рисунок 1: Главная страница разработанного приложения

Import Humans (JSON)

Select JSON file

Выберите файл

Файл не выбран

Upload and Import

Back

Expected format: array of objects with coordinates and car in the same record.

[{"name": "Alice", "mood": "SADNESS", "soundtrack": "track-001", "impactSpeed": 0.5, "weaponType": "AXE", "realHero": true, "hasToothpick": false, "coordinates": {"x": 10, "y": 20},

Import History

ID	Status	Imported	Started	Completed	File	Error
21	FAILED		2025-10-19 17:47:01	2025-10-19 17:47:01	humans.json	name already exists: Alice
20	FAILED		2025-10-19 17:47:01	2025-10-19 17:47:01	humans.json	name already exists: Alice
19	SUCCESS	2	2025-10-19 17:47:01	2025-10-19 17:47:01	humans.json	
16	SUCCESS	2	2025-10-19 17:46:12	2025-10-19 17:46:12	humans.json	

Рисунок 2: Страница импорта разработанного приложения

Исходный код
<https://github.com/senya-2011/is-course>

UML-диаграммы

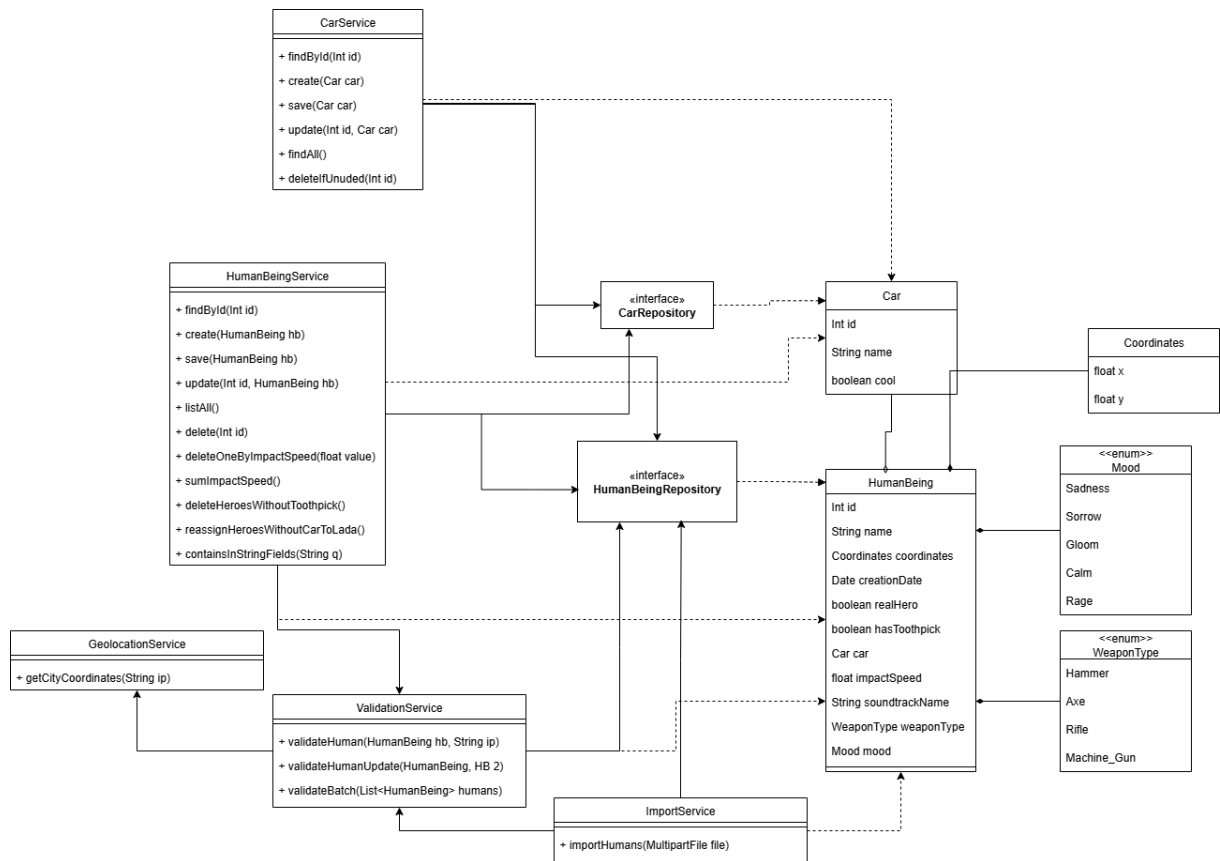


Рисунок 3: UML-диаграмма классов

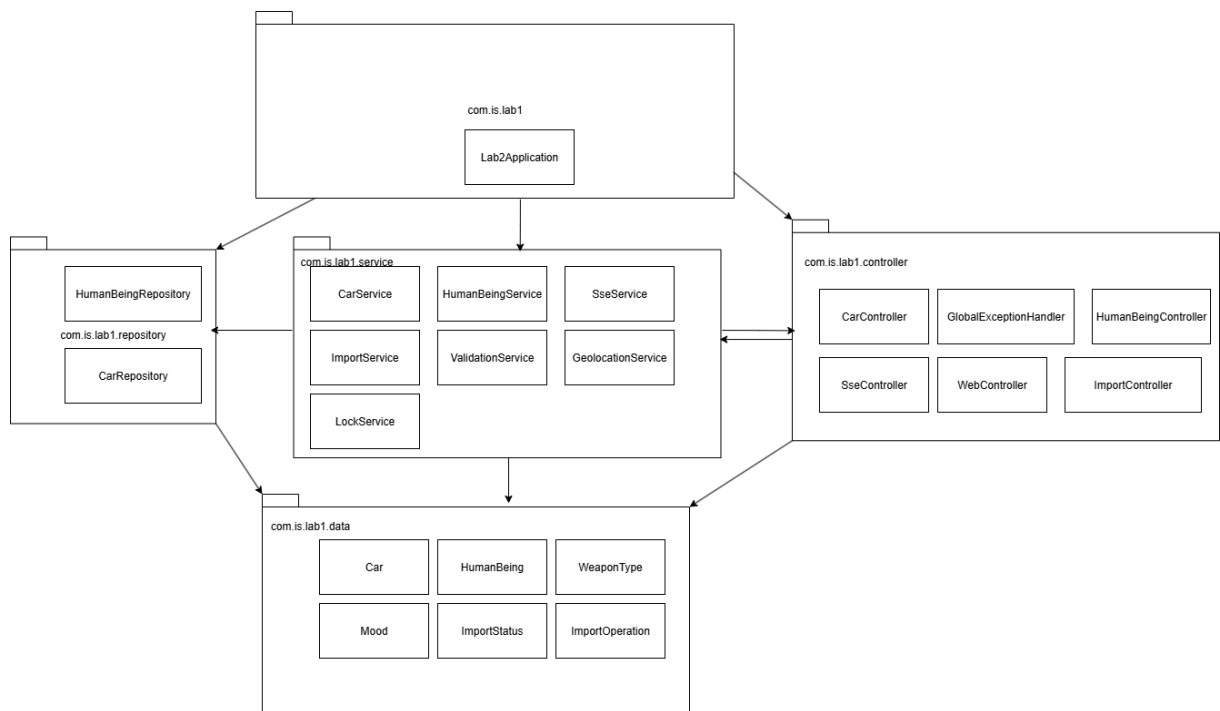


Рисунок 4: UML-диаграмма пакетов разработанного приложения

Выводы по работе

В ходе выполнения лабораторной работы была успешно реализована система массового импорта объектов. Добавлены новые бизнес-правила уникальности (ограничение владельцев автомобиля, запрет дублирования имен/координат/саундтреков, проверка координат города пользователя, временные ограничения для настройки RAGE), которые реализованы на уровне приложения с использованием PostgreSQL advisory locks для предотвращения race conditions. Реализованы блокировки на уровне транзакций и централизованная валидация с учетом исключения текущей записи при обновлении. JMeter тестирование подтвердило, что при одновременных операциях создания/обновления с одинаковыми уникальными значениями только одна операция выполняется успешно, остальные получают соответствующие ошибки валидации.