НИТУ МИСИС
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОМПЬЮТЕРНЫХ НАУК

Мальцев Никита Алексеевич

Тема:"Разработка клиент-серверного CRUD приложения."

КУРСОВАЯ РАБОТА ПО КУРСУ "ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ"

Научный руководитель:
преподаватель, к.т.н.
Максим Евгеньевич Козлов

Москва, 2023

Оглавление

1. Введение

Краткий обзор области

CRUD — акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание (англ. create), чтение (read), модификация (update), удаление (delete). Введён Джеймсом Мартином (англ. James Martin) в 1983 году как стандартная классификация функций по манипуляции данными.

В SQL этим функциям, операциям соответствуют операторы Insert (создание записей), Select (чтение записей), Update (редактирование записей), Delete (удаление записей). В некоторых CASE-средствах использовались специализированные CRUD-матрицы или CRUD-диаграммы, в которых для каждой сущности указывалось, какие базовые функции с этой сущностью выполняет тот или иной процесс или та или иная роль. В системах, реализующих доступ к базе данных через API в стиле REST, эти функции реализуются зачастую (но не обязательно) через HTTP-методы PUT, POST, GET, PATCH, DELETE.

Хотя традиционно оперирование в стиле CRUD применяется к базам данных, такой подход может быть распространён на любые хранимые вычислительные сущности (файлы, структуры в памяти, объекты). Шаблон проектирования ActiveRecord обеспечивает соответствие функций CRUD объектно-ориентированному подходу, и широко используется в различных фреймворках для доступа к базам данных из объектно-ориентированных языков программирования.

Клиент-сервер (англ. Client-server) — сетевая архитектура, в которой устройства являются либо клиентами, либо серверами. Клиентом (front end) является запрашивающая машина (обычно ПК), сервером (back end) — машина, которая отвечает на запрос. Оба термина (клиент и сервер) могут применяться как к физическим устройствам, так и к программному обеспечению.

Постановка задачи

Целью курсовой работы является разработка клиент-серверного приложения с использованием базы данных.

2. Теоретическая часть

2.1. Язык программирования Java

Для решения данной задачи использовались знания из области базовых средств объектно-ориентированного языка программирования Java. Понятие «объектно- ориентированный» относится к способу написания структурного кода Java, а именно: разделение кода на так называемые «классы», которые запускаются вместе, чтобы обеспечить согласованное порождение объектов. Такая структура программы приводит к универсальному и организованному коду, который легко редактировать и воспринимать. Кроме обычных методов классы могут определять специальные методы, которые называются конструкторами. Конструкторы вызываются при создании нового объекта данного класса и выполняют его инициализацию.

Главные достоинства данного языка – это его простота и объектно-ориентированный подход. Язык Java имеет чёткие синтаксические правила и понятную семантику. Для выполнения курсовой работы используется Java 17.

2.2. Формат CSV

В курсовой работе по заданию необходимо работать с данными в csv файлах. CSV- файлы (файлы данных с разделителями-запятыми) — это файлы особого типа, в них данные хранятся не в столбцах, а разделенные запятыми. Предназначением данной технологии является представление табличных данных. При этом формат является универсальным. С помощью csv-файлов возможно представить таблицы любых баз данных, так как синтаксис этого формата максимально прост и будет понятен практически любому с ним столкнувшемуся человеку.

Данная технология преобразования таблиц широко применяется на практике, даже при наличии многих других инструментов, например xml-формата. CSV нашел свое применение, как у web-разработчиков, так и у разработчиков мобильных технологий.

2.3. Работа с базой данных PostgreSQL

В данной работе я буду подключаться к базе данных PostgreSQL.

PostgreSQL – это свободная объектно-реляционная система управления базами данных (СУБД). Данная СУБД отличается от других тем, что она обладает объектно- ориентированным функционалом, в том числе полной поддержкой концепта ACID (Atomicity, Consistency, Isolation, Durability).

PostgreSQL обладает многими возможностями, которые привлекают разработчиков, например:

- PostgreSQL - бесплатное ПО с открытым исходным кодом
- PostgreSQL - кроссплатформенное ПО, к нему существуют интерфейсы из всех современных языков программирования
- PostgreSQL хорошо масштабируется и обеспечивает высокую производительность
- PostgreSQL очень надежна, аварийное завершение случается редко
- Развитая функциональность позволяет гибко решать различные задачи
- Развитая экосистема клиентов и административных средств позволяет легко и быстро выполнять такие рутинные задачи, как описание объектов базы данных, экспорт и импорт данных, резервное копирование и восстановление базы
- PostgreSQL легко интегрируется с другими СУБД, что открывает возможность для гибкой реализации программных проектов

2.4. Фреймворк Spring Framework

Для разработки консольного приложения был выбран фреймворк Spring Framework. Spring Framework (или коротко Spring) — это универсальный фреймворк с открытым исходным кодом для Java-платформы. Особенность данного фреймворка заключается в том, что пользователю надо просто писать классы, а создает объекты классов и вызывает методы уже сам фреймворк. Чаще всего, классы имплементируют какие-то интерфейсы из фреймворка или наследуют какие-то классы из него, таким образом получая часть уже написанной за пользователя функциональности.

Можно сказать, что использование Spring упрощает работу и экономит время, так как большую часть написания кода берет на себя Spring.

Также работая с фреймворком Spring, будем обращаться к Spring Data JPA. Это библиотека, которая добавляет дополнительный уровень абстракции поверх ORM реализации JPA. По умолчанию Spring Data JPA использует Hibernate, в качестве ORM провайдера, чтобы выполнять запросы. JpaRepository предоставляет связанные с JPA методы, такие как очистка контекста сохранения и удаление записей в пакете.

2.5. Библиотека Hibernate

Hibernate — это библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения (ORM), самая популярная реализация спецификации JPA. Позволяет сократить объёмы низкоуровневого программирования при работе с реляционными базами данных; может использоваться как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с существующей базой.

Библиотека не только решает задачу связи классов Java с таблицами базы данных, но и также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL- и JDBC- кода. Hibernate автоматизирует генерацию SQL-запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос приложения на любые базы данных SQL.

3. Практическая часть

3.1. Детальное требование к программе

Обязательный функционал для варианта 18

- Работа с данными в csv файлах
- Автоматическое создание таблиц в базе данных
- Веб-страница для загрузки данных в таблицы базы данных из csv файлов
- Вывод результатов в csv файлы
- Выделение дня транзакции в отдельную колонку в таблице транзакций
- Веб-страница вывода средних значений по полю amount, сгруппировав данные по дню, а в рамках дня по MCC коду со следующими условиями: MCC коды, встречаются в выборке более чем 3 раза

Дополнительный функционал

- Создание сессии пользователя с возможностью регистрации и дальнейшей повторной инициализации сессии
- Страница регистрации
- Страница инициализации

3.2. Декомпозиция технического задания

Разработка приложения осуществлялась на основе паттерна MVC. Соответсвенно, первый этап декомпозиции - это уровни моделей, контроллеров и представления.

Дальнейшая декомпозиция осуществляется опционально внутри данных блоков. В действительности, реализация той или иной функциональности проходит сквозь данные блоки, поэтому итоговая декомпозиция производилась именно по таким "логическим блокам"реализации той или иной фунциональности приложения.

Итоговые этапы разработки можно детально отследить по истории git-коммитов приложения на данных платформах:

- https://edu.gitflic.ru/project/malcevnik99/transaction-application
- https://github.com/mmmhdp/transaction_application

3.3. Разработка

Итоговая структура проекта

Итоговая структура проекта представлена на данных изображениях 3 и 4.

Код

Код приложения хранится в облачных репозиториях:

- на GitHub в общем доступе по ссылке: https://github.com/mmmhdp/transaction_application.
- на GitFlic в учебной части: https://edu.gitflic.ru/project/malcevnik99/transaction-application

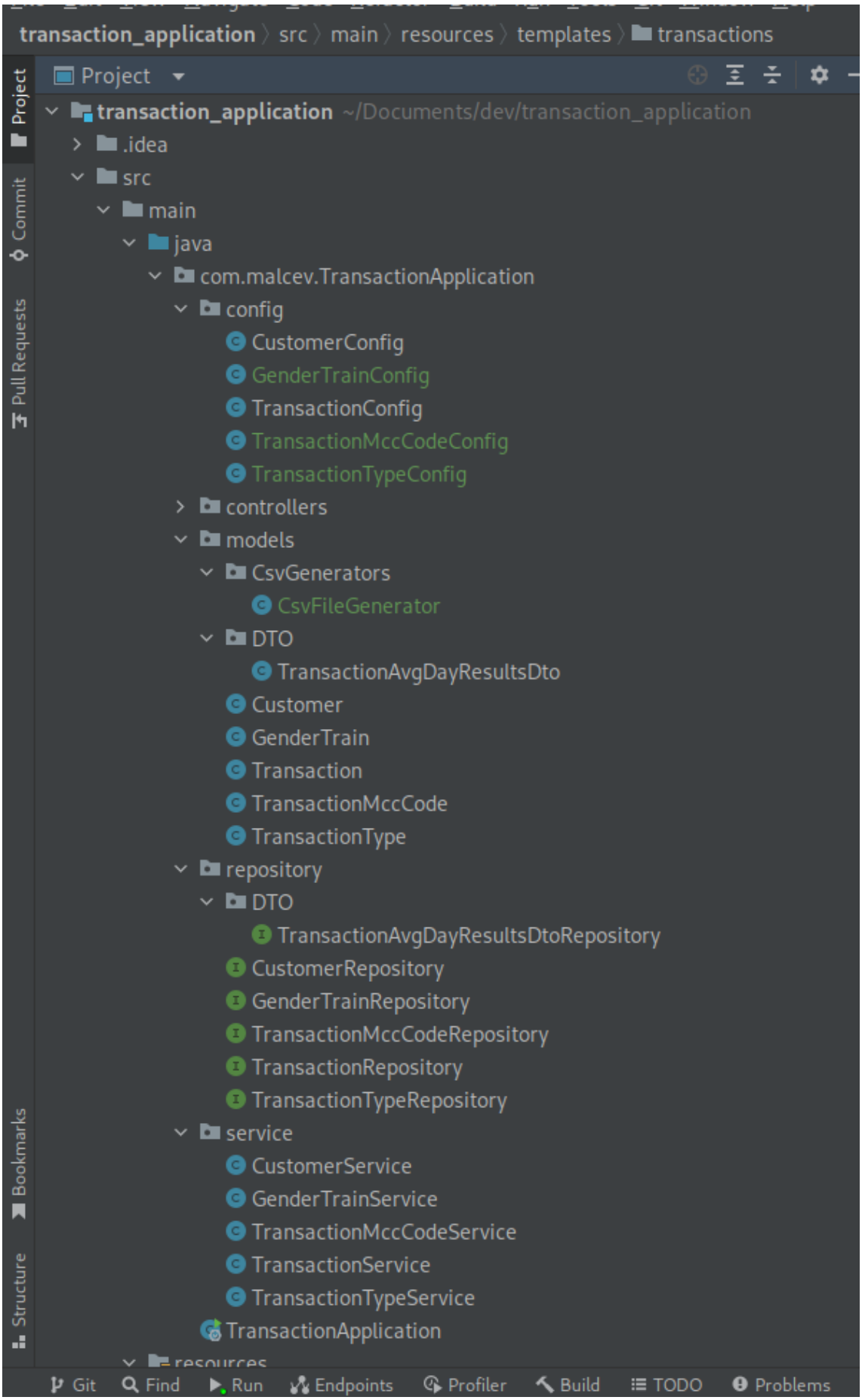


Рис. 1. Структура приложения

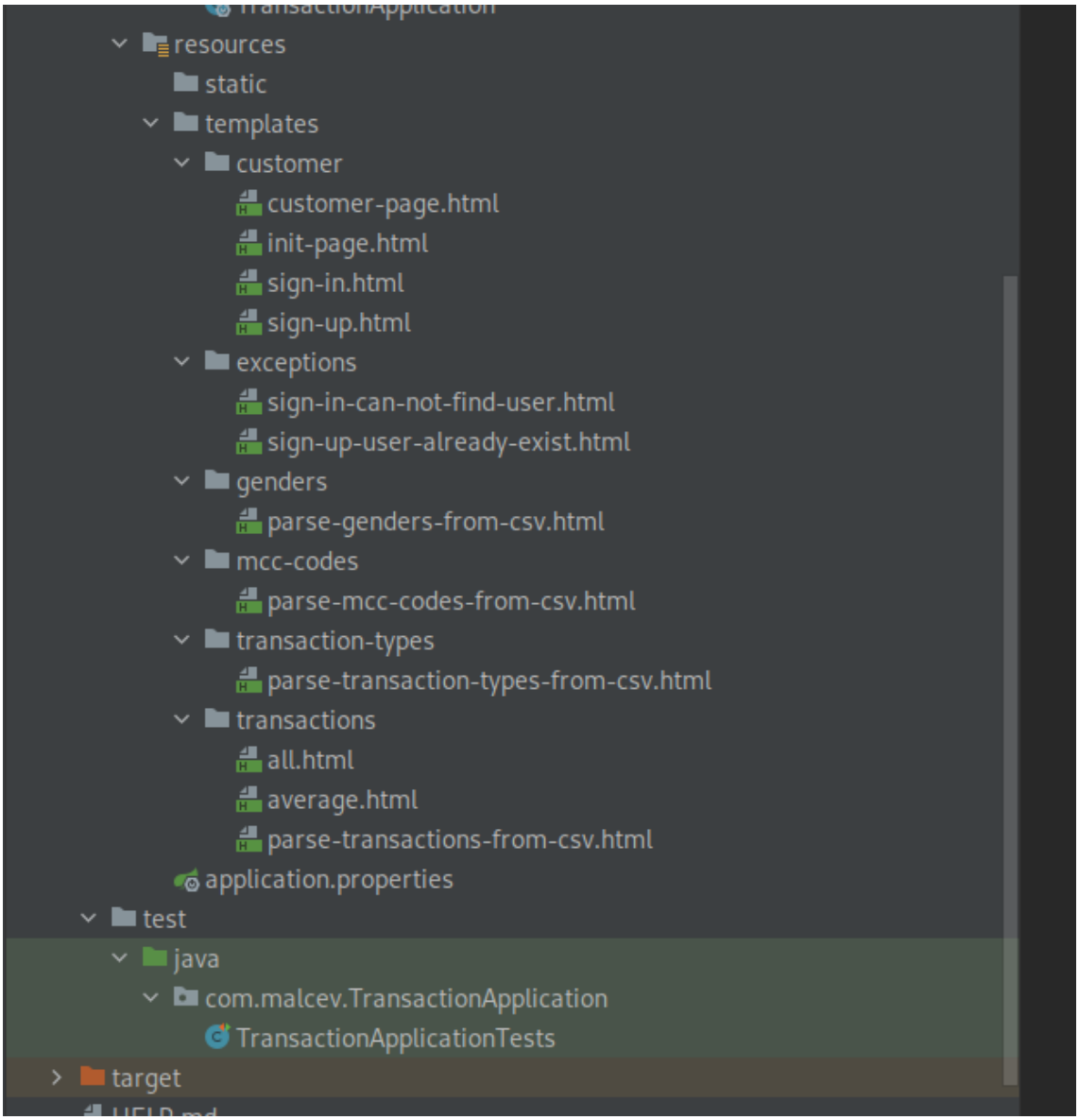
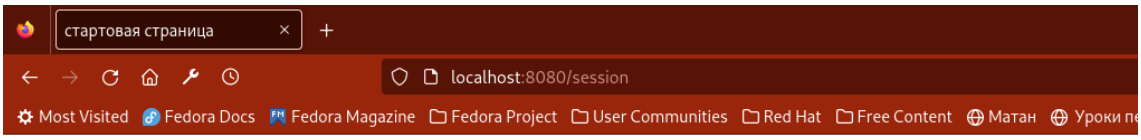


Рис. 2. Структура приложения

3.4. Результат работы

Работу приложения можно оценить через работу с его функционалом. Визуализация ниже, действия происходят одно за другим в указанной последовательности.



добро пожаловать в наше приложения для ведения учёта транзакций

выберите способ инициализации сессии
войти в существующий аккаунт
зарегистрировать новый аккаунт

Рис. 3. Стартовая страница приложения

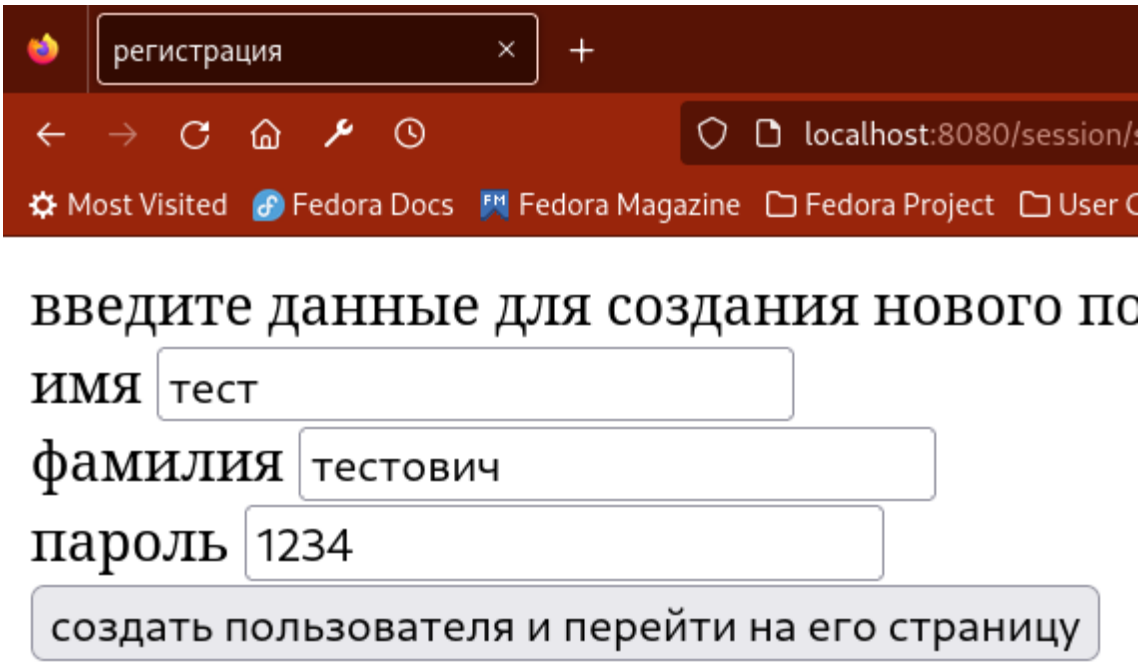


Рис. 4. Регистрация

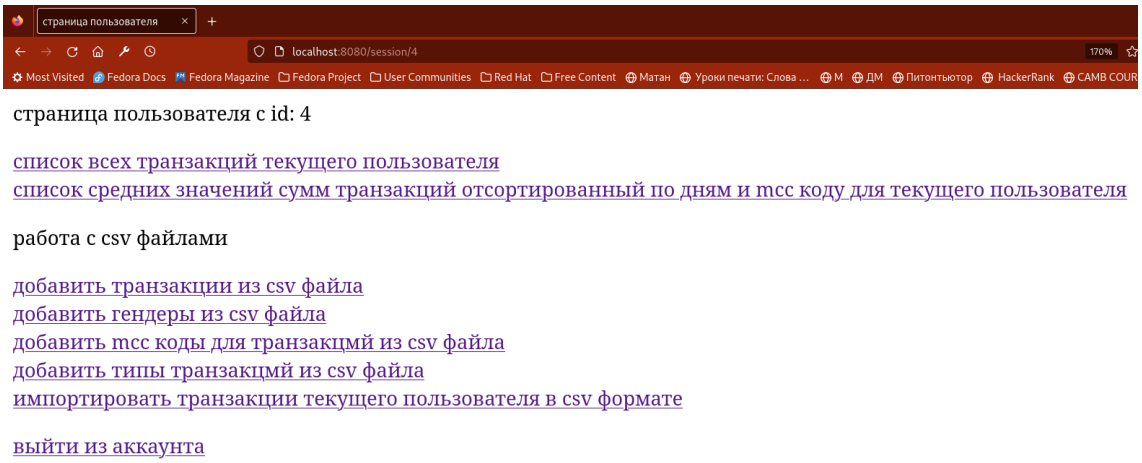


Рис. 5. Страница пользователя после регистрации пользователя

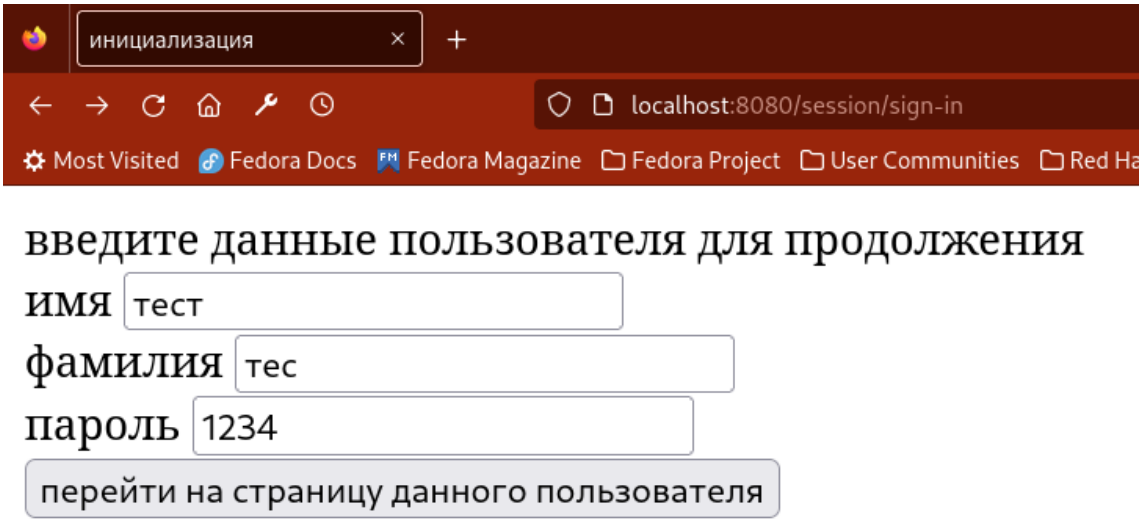


Рис. 6. Пример неудачной попытки войти с неверными данными пользователя

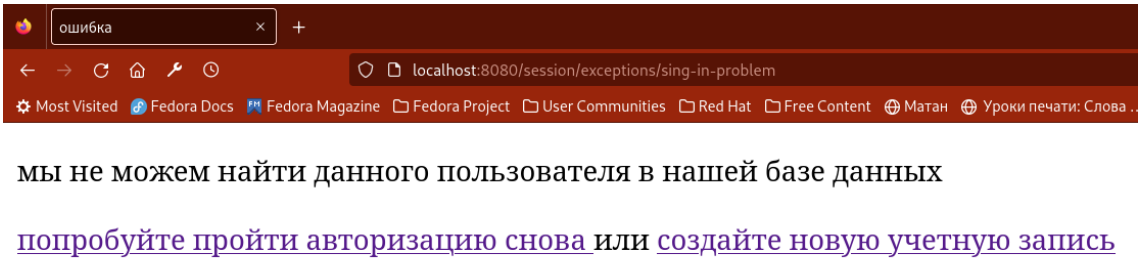


Рис. 7. Результат неудачной инициализации

страница пользователя с id: 4

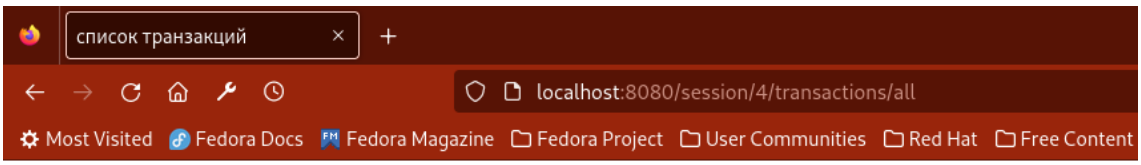
список всех транзакций текущего пользователя
список средних значений сумм транзакций отсортированный по дням и mcc коду для текущего пользователя

работа с csv файлами

добавить транзакции из csv файла
добавить гендеры из csv файла
добавить mcc коды для транзакцмй из csv файла
добавить типы транзакцмй из csv файла
импортировать транзакции текущего пользователя в csv формате

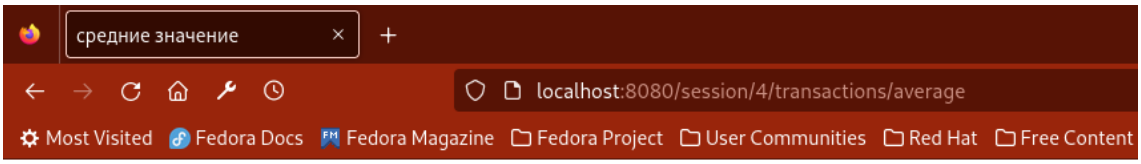
выйти из аккаунта

Рис. 8. Результат удачной инициализации



Список транзакций пользователя:

Рис. 9. До считывания данных



Список средних значений:

Рис. 10. До считывания данных

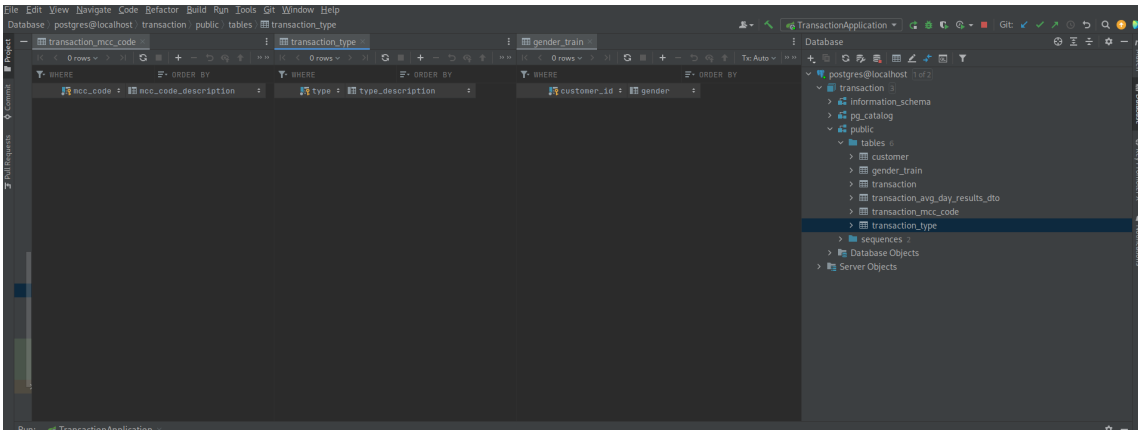


Рис. 11. База данных до считывания данных

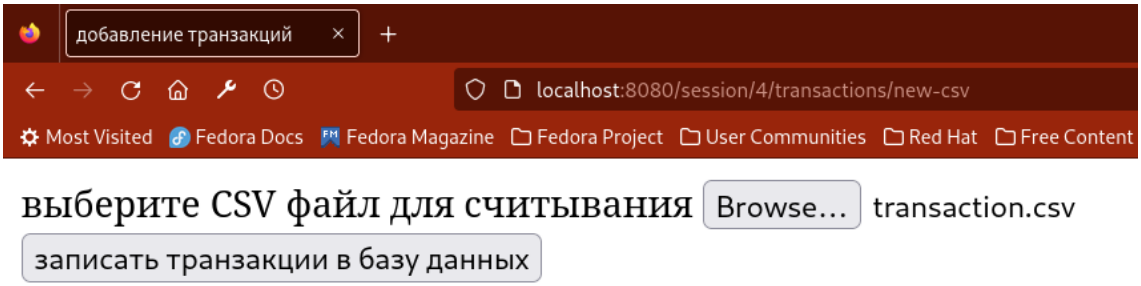


Рис. 12. Добавления списка транзакций из csv файла

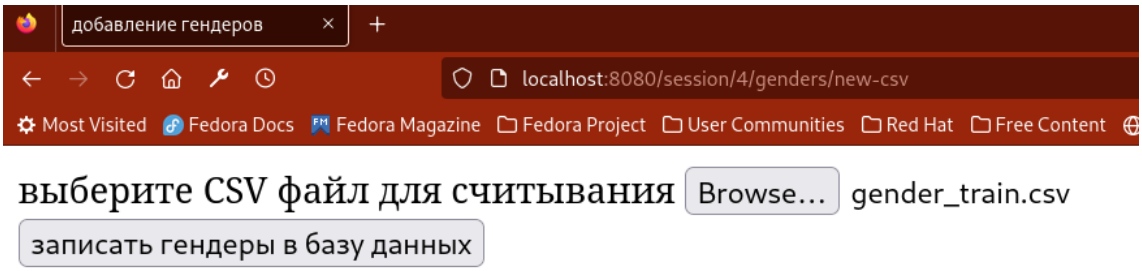


Рис. 13. Добавления списка гендеров пользователей из csv файла

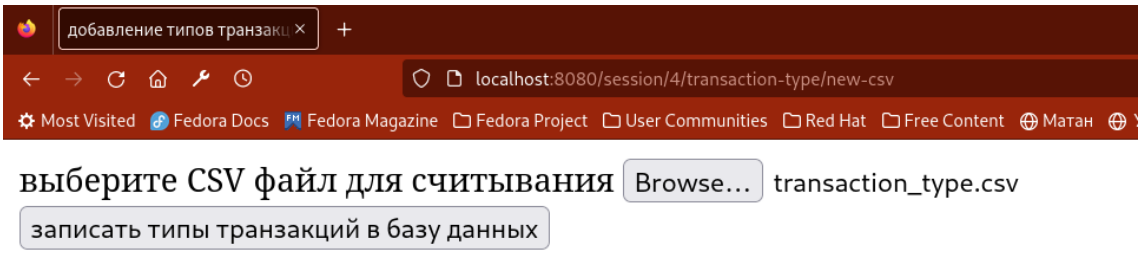


Рис. 14. Добавления списка типов транзакций из csv файла

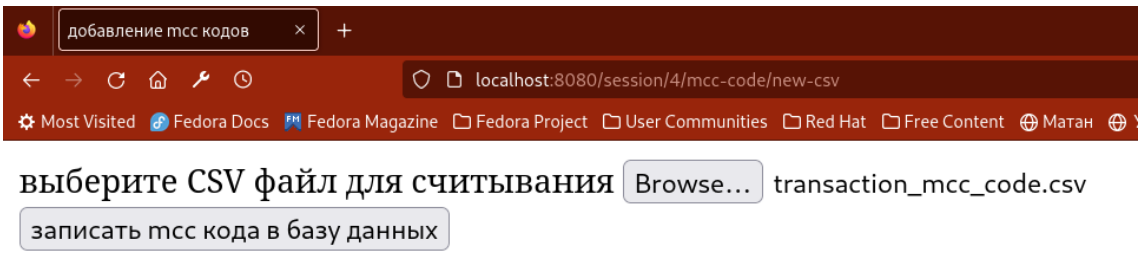


Рис. 15. Добавления списка mcc кодов для транзакций из csv файла

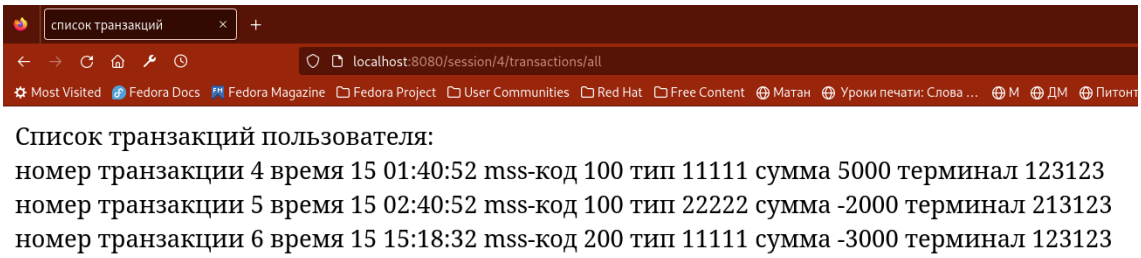


Рис. 16. Список транзакций пользователя после считывания данных

Рис. 17. Специальный список после считывания данных

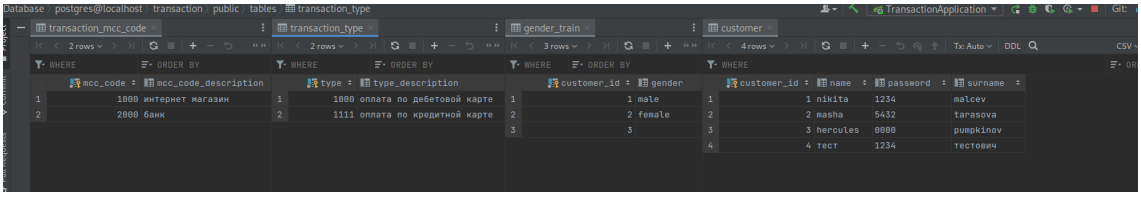
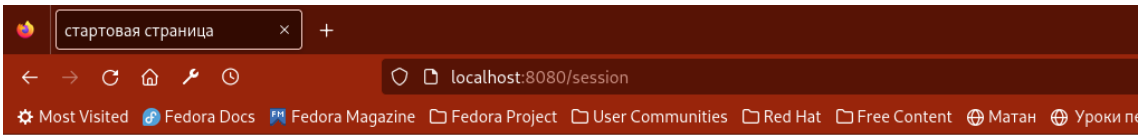


Рис. 18. Состояние базы данных после считывания данных



добро пожаловать в наше приложения для ведения учёта транзакций

выберите способ инициализации сессии
войти в существующий аккаунт
зарегистрировать новый аккаунт

Рис. 19. Результат выхода из авторизации

4. Заключение

В результате выполнения курсовой работы была изучена специальная литература, описаны теоретические аспекты и раскрыты ключевые понятия исследования, а самое главное было создано клиент-серверное приложение с использованием системы управления базами данных PostgreSQL. Данное приложение соответствует требованиям задания. При создании приложения были использованы и изучены такие инструменты, как язык программирования Java, система управления базами данных PostgreSQL и фреймворк Spring Boot и часть его библиотек.

5. Список источников и литературы

- Spring Quickstart Guide [электронный ресурс]: https://spring.io/quickstart
- Работаем с файлами CSV в Java с использованием библиотеки OpenCSV [электронный ресурс]: https://techcave.ru/posts/99-rabotaem-s-failami-csv-v..
- Thymeleaf [электронный ресурс]: https://www.thymeleaf.org/documentati
- С. А. Орлов «Программная инженерия. Технологии разработки программного обеспечения». 5-е издание обновленное и дополненное – СПб.: Питер, 2016–640 с.
- К. Сьерра «Изучаем Java» - Эксмо, 2012–720 с.
- Г. Шилдт «Java. Полное руководство» 10-е издание - Диалектика-Вильямс, 2018– 1488 с.
- Ю. Козмина, Р. Харроп «Spring 5 для профессионалов» - Диалектика-Вильямс, 2019– 1120 с.
- А. Швец «Погружение в паттерны проектирования» - Самиздат, 2018–406 с.