|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: «Северная часть веб-приложения «Онлайн банк»»

Студент: Магин Константин Андреевич

Группа: ИКБО-10-19

Работа представлена к защите\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/ Магин К. А. /

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ст. преподаватель Синицын А. В.

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/Синицын А. В./

Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине: Разработка серверных частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Магин Константин Андреевич

Группа: ИКБО-10-19

Срок представления к защите: 23.11.2021

Руководитель: старший преподаватель Лобанов А. А.

**Тема** «Северная часть веб-приложения «Онлайн банк»»

**Исходные данные**: используемые технологии: HTML5, CSS3, Java/Kotlin/PHP (на выбор), Jetbrains Intellij IDEA/Jetbrains PHPStorm/Eclipse (на выбор), SQL/noSQL СУБД (на выбор), наличие: межстраничной навигации, внешнего вида страниц, соответствующего современным стандартам веб-разработки, использование паттерна проектирования (MVC, Clear Architecture, DDD). Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала**:

1. Провести анализ предметной области разрабатываемого веб-приложения. 2. Обосновать выбор технологий разработки веб-приложения. 3. Разработать архитектуру веб-приложения на основе выбранного паттерна проектирования. 4. Реализовать слой серверной логики веб-приложения с применением выбранной технологии. 5. Реализовать слой логики базы данных. 6. Разработать слой клиентского представления веб-приложения 7. Создать презентацию по выполненной курсовой работе.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: \_\_\_\_\_\_\_\_\_\_\_/Р. Г. Болбаков/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А. А. Лобанов/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР получил: \_\_\_\_\_\_\_\_\_\_\_/А. К. Магин/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

УДК 004.4

Магин К. А.Приложение «Онлайн банк»/ **Курсовая работа** по дисциплине «Шаблоны программных платформ на языке Джава» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04 «Программная инженерия» (5-ый семестр) / руководитель старший преподаватель Синицын А. В. / кафедра ИППО Института ИТ МИРЭА – с. 26, табл. 3, ист. 6.

Целью работы является разработка приложения «Онлайн банк».

В рамках работы осуществлены краткий анализ аналогов приложения, разработка архитектуры базы данных, анализ разработки программного продукта, контейнеризация и тестирование.

Magin K. A. Appendix "Barbershop" / Course work on the discipline "Patterns of software platforms in Java language" profile "Development of software products and design of information systems" direction of professional training for bachelor's degree 09.03.04 "Software engineering" (4th semester) / head senior lecturer Sinitsyn A.V. / Department of IOPS of the Institute of IT MIREA - p. 26, tab. 3, ist. 6.

The aim of the work is to develop the Barbershop application.

As part of the work, a brief analysis of the analogs of the application, the development of the database architecture, the analysis of the development of a software product, containerization and testing were carried out.

М. МИРЭА. Ин-т ИТ. Каф. ИиППО. 2021 г. Магин К. А.

**ГЛОССАРИЙ**

ФИО — фамилия, имя, отчество;

БД — база данных;

СУБД — система управления базами данных.

REST (Representational state transfer) – это стиль архитектуры программного обеспечения для распределенных систем.

HTTP (HyperText Transfer Protocol) – протокол передачи гипертекста.

IDE (Integrated Development Environment) – интегрированная среда разработки приложений.

XML — расширяемый язык разметки.

HTML – стандартизированный язык разметки документов во Всемирной паутине.

XHTML – семейство языков разметки веб-страниц на основе XML, повторяющих и расширяющих возможности HTML.

JPA — спецификация API Java EE, предоставляет возможность сохранять в удобном виде Java-объекты в базе данных.

MVC – паттерн Model — View — Controller (Модель — Отображение (далее — Вид) — Контроллер). Паттерн MVC разделяет аспекты приложения (логику ввода, бизнес-логику и логику UI), обеспечивая при этом свободную связь между ними.

CBS (core banking system) - комплекс программного и технического обеспечения, направленный на автоматизацию банковской деятельности.

НМТ – номер мобильного телефона

C2A – перевод с карты на счет

A2A – перевод с счета на счет

A2C – перевод с счета на карту

C2C – перевод с карты на карту

**АННОТАЦИЯ**

Курсовая работа содержит \_ страниц отчёта, \_ иллюстраций, \_ использованных источников информации.

Целью данной курсовой работы является подробное описание процесса создания и разработки серверного программного приложения. Работа выполнена с использованием языка программирования Java, и фреймворком Spring, в частности Spring Data JPA, Spring MVC, Spring Security. В качестве системы управления базами данных использовалась PostgreSQL.

Результатом данной курсовой работы является созданное серверное программное приложение.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 6](#_Toc87975711)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc87975712)

[Выводы к разделу 10](#_Toc87975713)

[2. ВЫБОР И ОБОСНОВАНИЕ ТЕХНОЛОГИЙ 11](#_Toc87975714)

[Выводы к разделу 12](#_Toc87975715)

[3. АРХИТЕКТУРА ПРИЛОЖЕНИЯ НА ОСНОВЕ ВЫБРАННОГО ПАТТЕРНА 13](#_Toc87975716)

[Выводы к разделу 18](#_Toc87975717)

[4. РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ИНТЕРНЕТ-РЕСУРСА 20](#_Toc87975718)

[Выводы к разделу 21](#_Toc87975719)

[4.1 Проектирование структуры базы данных 22](#_Toc87975720)

[4.2 Тестирование приложения 23](#_Toc87975721)

[Выводы к разделу 27](#_Toc87975722)

[ЗАКЛЮЧЕНИЕ 28](#_Toc87975723)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc87975724)

[ПРИЛОЖЕНИЕ 30](#_Toc87975725)

[Таблицы прецедентов 30](#_Toc87975726)

# ВВЕДЕНИЕ

Постоянное развитие современных технологий и их повсеместное распространение, привело к развитию и автоматизации процессов в бизнес-сегменте. В частности, это затронуло и банковский сектор. Создание автоматизированных банковских систем для обработки транзакций, систем процессинга банковских карт, сервисов предоставления различных услуг конечным пользователям (в т.ч. удаленных каналах самообслуживания) — это технологическая основа банковского бизнеса на сегодня. Удаленное обслуживание на сегодняшний день стало стандартом, что послужило причиной выбора темы курсовой.

Выбор темы курсовой работы обусловлен интересом к разработке полноценной серверной части с различным функционалом, который мог бы иметь практическое применение и позволил бы получить навыки использования фреймворка для языка Java Spring и других средств разработки. Северная часть веб-приложения «Онлайн банк» не простая в реализации задача, помогает отточить навыки построения клиент-серверных решений (архитектура приложения и корректное взаимодействие всех её компонентов).

Приложение «Онлайн банк» является одним из составляющих ИТ-ландшафта банка. Данной приложение выступает в роли посредника между CBS и конечными пользователями-клиентами банка. С его помощью пользователь получает все возможные банковские услуги, сервисы.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметной областью для данной курсовой работы является исследование приложений в сфере онлайн банкинга. Был произведен анализ предметной области, в результате которого были сформированы основные требования к приложению.

Онлайн банк — это приложение, занимающееся предоставлением услуг конкретного банка для авторизированных пользователей. Используя такое приложение, клиенты банка могут получать продукты, услуги и сервисы без посещения офисов банка. Данный режим обслуживания позволяет банкам:

* снизить расходы на обслуживание филиальной сети (или вообще от неё отказаться, например, как Тинькофф банк);
* Расширить время обслуживания клиентов до 24 часов в сутки и получать от этого дополнительную прибыль.

Приложение онлайн банка является одним из составляющих большого ИТ-ландшафта банка, включающего в себя системы CBS.

Приложения в данной сфере обычно представляют собой мобильное или интернет веб-приложение.

На рисунке 1.1 изображён интерфейс приложения «ВТБ-онлайн».

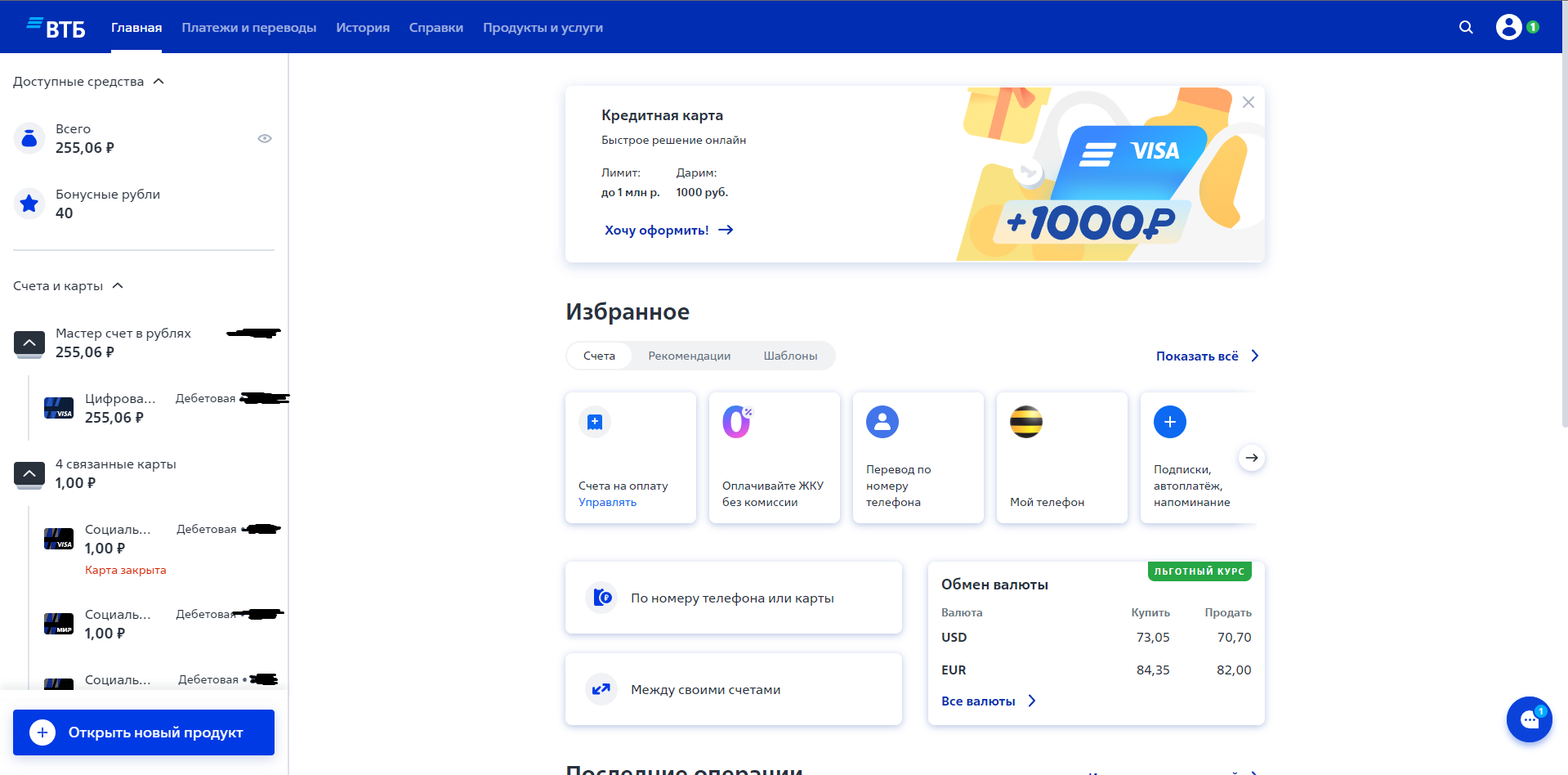


Рисунок 1.1 - Интерфейс приложения «ВТБ»

На рисунке 1.1 изображён интерфейс приложения «Тинькофф».

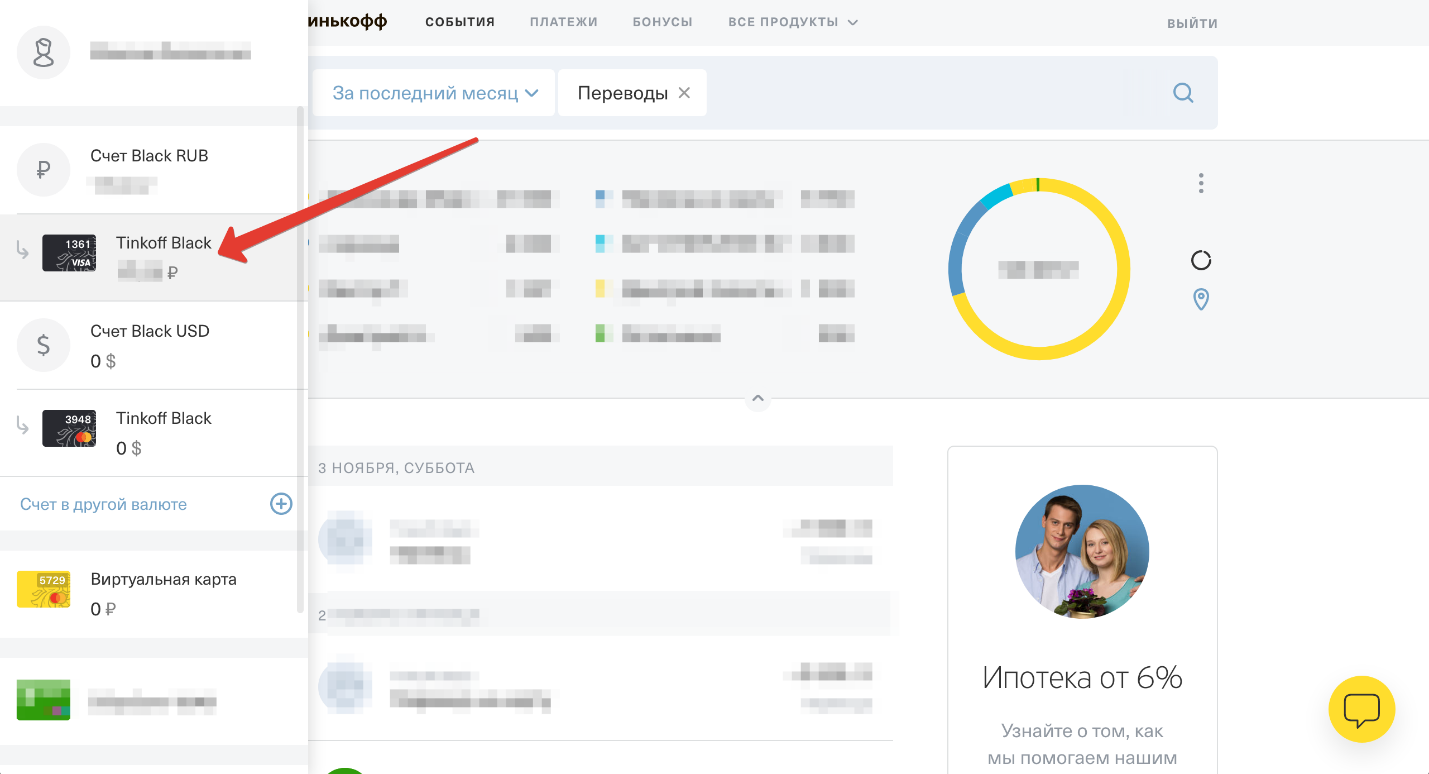


Рисунок 1.2 - Интерфейс приложения «Тинькофф»

«ВТБ-онлайн» и «Тинькофф» являются примерами таких систем.

Основными преимущества приложения для конечного пользователя являются:

1. Простота использования;
2. Информативность;
3. Оперативность;
4. Безопасность;
5. Количество и качество предоставляемых услуг.

Приложения «ВТБ-онлайн» и «Тинькофф» - одни из самых лучших в своей сфере. Использование передовых систем и технологий позволяют ВТБ и Тинькофф оставаться одними из лидеров на рынке онлайн банкинга.

Лучшем же является Тинькофф согласно аналитическому агентству Markswebb [7].

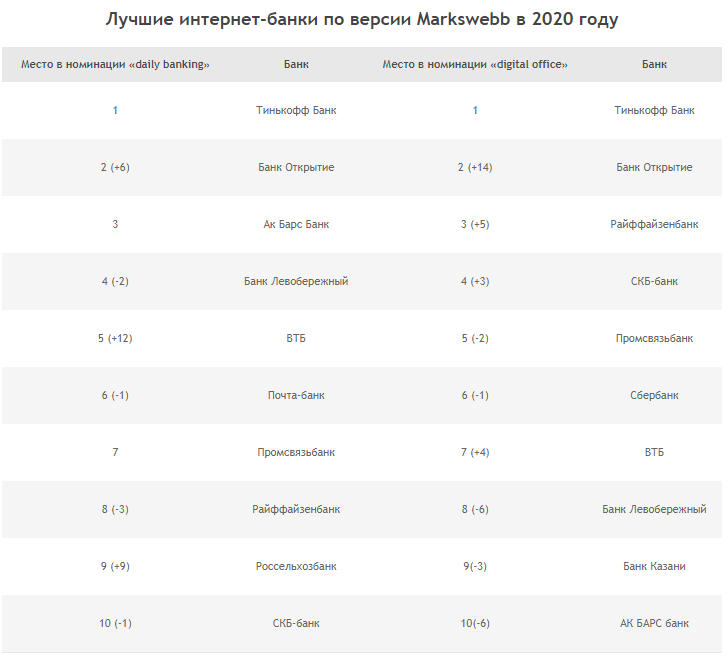


Рисунок 1.3 – рейтинг интернет-банков

Данный веб-ресурс можно использовать как эталон для разработки приложения в данной работе, так как является хорошим примером ресурса с подобным функционалом.

При проектировании приложения, были приняты ограничения, связанные с типом разрабатываемого приложения:

1. Приложения не является системой CBS;
2. Данные приложение получает из внешней системы CBS;
3. Приложение реализует только функционал онлайн банка;
4. Приложение имеет ограниченный функционал из-за отсутствия доступной полнофункциональной системы CBS в данной курсовой работе.

# Выводы к разделу

Проведя анализ предметной области, можно сделать вывод, что базовый функционал приложения «Онлайн банк» заключается в возможности регистрации с получением данных о счете пользователя, а также о картах пользователя и его транзакциях.

Потому разработка данных функций должна иметь наивысший приоритет.

Учитывая описанные выше ограничения, были сформированы функциональные требования, которым должно отвечать разработанное приложение:

1. Функции регистрации;
2. Функции идентификации, аутентификации, авторизации;
3. Функции контроля полномочий пользователя на основе ролевой модели;
4. Функции получения информации о состоянии клиентских счетов;
5. Функции получения информации о картах пользователя;
6. Функции получения информации о транзакциях/ операциях пользователя;

# ВЫБОР И ОБОСНОВАНИЕ ТЕХНОЛОГИЙ

Для функционирования серверной части веб-приложения использовались следующие технологии:

1. Spring Framework (Spring Core)- фреймворков для создания веб-приложений на Java. Он обеспечивает комплексную модель разработки и конфигурации для современных бизнес-приложений на любых платформах.
2. Spring Boot – фреймворк, упрощает создание Spring приложений и обеспечивает такими возможностями, как мониторинг состояниями и расширенная конфигурация.
3. Spring Security – фреймворк, предоставляющий механизмы создания аутентификации, авторизации и других возможностей обеспечения безопасности для приложения, созданного с использованием Spring Framework.
4. Spring Data JPA – фреймворк, предоставляющий инструменты и механизмы работы с базой данных посредством JPA репозиториев.
5. Spring MVC – фрейворк, предоставляющий инструменты и механизмы для реализации архитектуры паттерна MVC при помощи слабо связанных готовых компонентов.
6. JUnit – библиотека для модульного тестирования приложений на языке Java.
7. Java – строго типизированный объектно-ориентированный язык программирования общего назначения, одной из областей применения является разработка серверных приложений.
8. PostgreSQL – реляционная СУБД.
9. Maven - фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM, являющемся подмножеством XML.
10. Thymeleaf — это шаблонизатор Java XML / XHTML / HTML5, который может работать как в веб-средах, так и не в веб-средах. Он лучше подходит для обслуживания XHTML / HTML5 на уровне представления веб-приложений на основе MVC, но он может обрабатывать любой XML-файл даже в автономных средах. Он обеспечивает полную интеграцию с Spring Framework.
11. Bootstrap – фреймворк, предоставляющий инструменты и механизмы для ускоренного создания веб-страниц.

Все вышеперечисленные технологии хорошо себя зарекомендовали для разработки онлайн приложений и используются во множестве систем, включая интернет-банки. Выбранные технологии позволяют вести разработку промышленных приложений, получая наилучший результат и качество за кратчайшие сроки, что является критичным для получения успешных результатов в рамках курсовой работы.

Для быстрой и удобной разработки использовался следующие инструменты:

Intellij idea – IDE для многих языков программирования, в частности JavaScript, разработанная компанией JetBrains.

# Выводы к разделу

Стэк технологий должен отвечать следующим требованиям:

1. Возможность разработки приложения в кратчайшие сроки;
2. Возможность реализации MVC паттернов;
3. Возможность обеспечения безопасности приложения;
4. Возможность обработки данных реляционных бах данных приложением.

Вышеперечисленные преимущества необходимы для достижения наилучшего результата при разработке приложения в рамках курсовой работы.

В результате были выбраны технологии, соответствующие данным требованиям.

# АРХИТЕКТУРА ПРИЛОЖЕНИЯ НА ОСНОВЕ ВЫБРАННОГО ПАТТЕРНА

После формирования списка требований в анализе предметной области становится возможным сформулировать список конкретных задач, которым должно отвечать разработанное приложение.

Для отображения списка задач была использована техника «Пользовательских историй», одному из способов описания требований к системе.

На таблицах 3.1-3.7 отображены задачи в виде пользовательских историй:

Таблица 3.1 – Пользовательская история о регистрации клиента

|  |  |  |
| --- | --- | --- |
| Заголовок | | Регистрация |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Зарегистрироваться |
| Цель | Чтобы | Подключить услугу онлайн банка |

Таблица 3.2 – Пользовательская история о получении информации о счете клиента.

|  |  |  |
| --- | --- | --- |
| Заголовок | | Запись на прием |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Узнать остаток на счете |
| Цель | Чтобы | Посмотреть количество своих средств |

Таблица 3.3 – Пользовательская история о входе в приложение клиентом.

|  |  |  |
| --- | --- | --- |
| Заголовок | | Запись на прием |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Войти в приложение |
| Цель | Чтобы | Посмотреть информацию о счете |

Таблица 3.4 – Пользовательская история о получении информации о транзакциях клиента.

|  |  |  |
| --- | --- | --- |
| Заголовок | | Запись на прием |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Получить список транзакций моего счета |
| Цель | Чтобы | Получить информацию о поступлениях и списаниях с моего счета |

Таблица 3.5 – Пользовательская история о получении информации о картах клиента.

|  |  |  |
| --- | --- | --- |
| Заголовок | | Запись на прием |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Получить список моих карт |
| Цель | Чтобы | Получить информацию об активных картах моего счета |

Таблица 3.6 – Пользовательская история о проведении транзакции.

|  |  |  |
| --- | --- | --- |
| Заголовок | | Запись на прием |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Провести транзакцию |
| Цель | Чтобы | Перевести/ пополнить карту/ счет |

Таблица 3.7 – Пользовательская история о выходе клиента из приложения.

|  |  |  |
| --- | --- | --- |
| Заголовок | | Запись на прием |
| Заказчик | Как | Пользователь |
| Примечание | Я хочу | Выйти из приложения |
| Цель | Чтобы | Закончить работу с приложением “Онлайн банк” |

Для получения более подробной информации см. таблицы прецедентов.

На основе списка задач, была разработана архитектура приложения на основе MVC паттерна. Согласно паттерну, состоящего из слоёв Model, View и Control, были созданы следующие компоненты системы, представленные на рисунке 3.1.

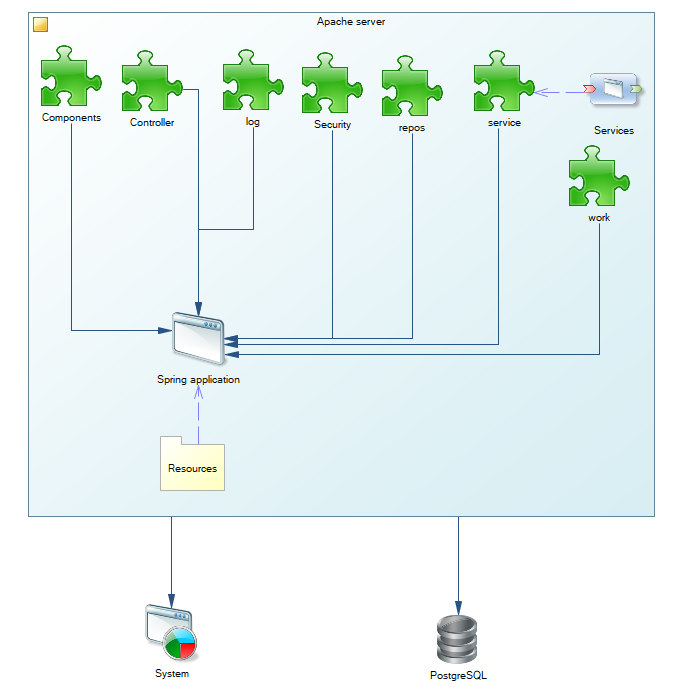


Рисунок 3.1 – архитектура приложения

Архитектура состоит из следующих компонентов:

1. Components
2. Controller
3. Log
4. Security
5. Repos
6. Service
7. Work
8. Resources
9. Spring application

В свою очередь, компоненты приложения реализуют конкретные слои паттерна MVC:

1. Компоненты Components, Log, Repos, Service реализуют model слой приложения. Основные функции компонентов заключаются в обработке данных, внесении и фиксировании их изменений в приложении.
2. Компоненты Controller, Work и Security реализуют control слой приложения, взаимодействующий с слоем model. Основные функции компонентов заключаются в интерпретации действий представления и оповещении приложения о необходимости изменений.
3. Компонент Resources реализует view слой приложения. Основные функции компонента заключаются в отображении данных приложения пользователю, реагируя на изменения приложения.

Согласно вышеописанной реализации паттерна MVC была создана диаграмма информационных потоков, которая описывает внутреннее взаимодействие компонентов системы:

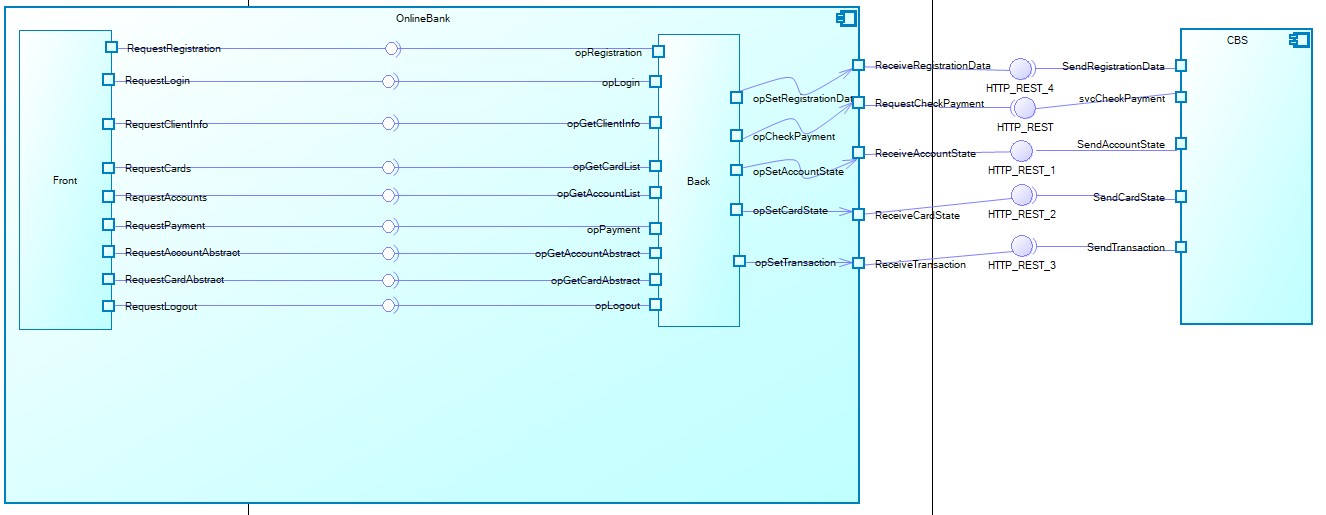


Рисунок 3.2 – схема информационных потоков приложения

Также была разработана архитектура системы в соответствии с паттерном 3-layer, представленным на рисунке 3.3.

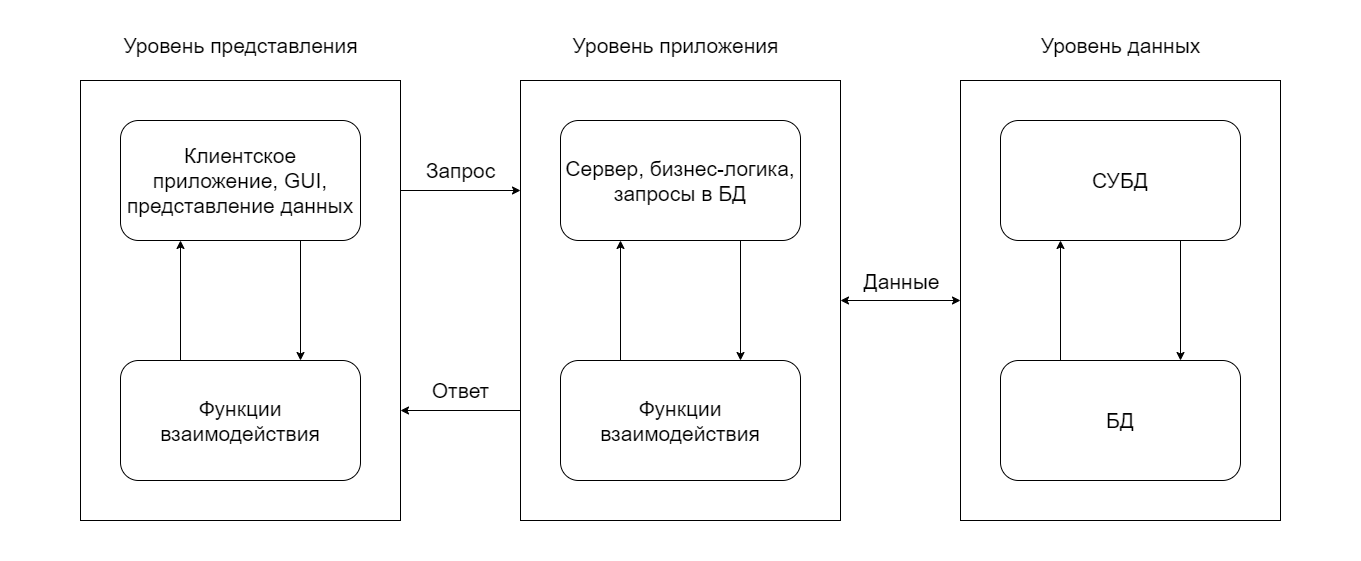


Рисунок 3.3 архитектура системы

Для дальнейшего развертывания приложения необходимо следовать схеме развертывания, представленной на рисунке 3.4:

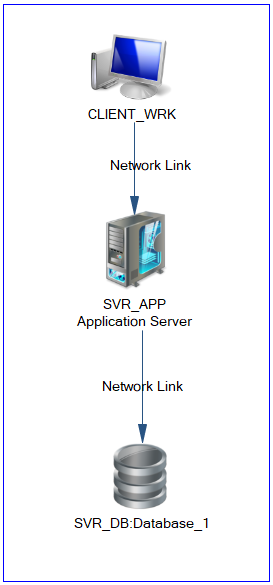


Рисунок 3.4 – схема развертывания приложения

# Проектирование структуры базы данных

При проектировании структуры базы данных было решено использовать следующий шаблон.

Исходя из темы курсовой работы были созданы основные таблицы: bank\_user, transaction, card\_transaction, roles, users\_roles, pay\_system, status, account, client, tariff.

Общая структура данных представлена на рисунках 3.1.1 – 3.1.2

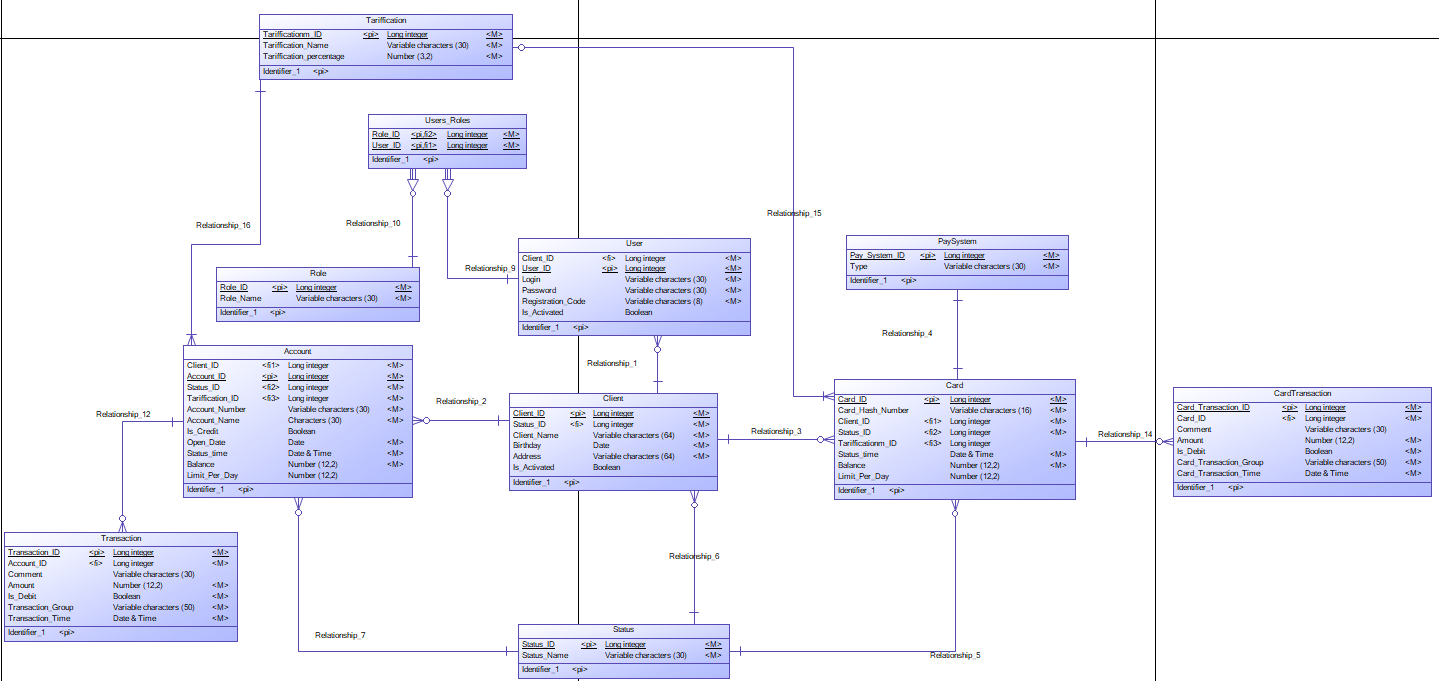


Рисунок 3.1.1 – логическая схема структуры базы данных

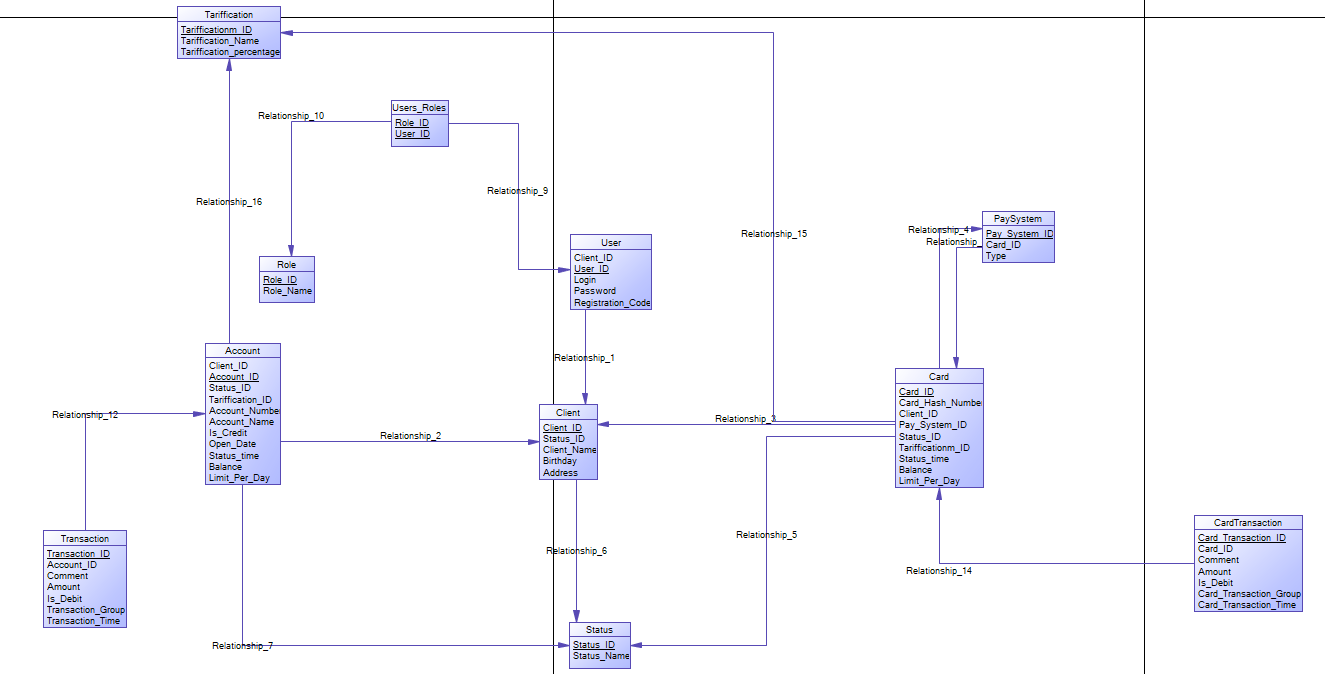


Рисунок 3.1.2 – физическая схема структуры базы данных

# Выводы к разделу

В результате разработки архитектуры приложения:

1. Разработана архитектура приложения, основанная на выбранном паттерне MVC;
2. Создана схема информационных потоков приложения;
3. Разработана архитектура системы в соответствии с паттерном 3 layer;
4. Создана схема дальнейшего развертывания системы.
5. Спроектирована структура базы данных

Данная архитектура позволяет реализовать приложения, которое удовлетворяет требованиям, которые были сформированы при анализе предметной области.

Согласно разработанной архитектуре, была произведена дальнейшая разработка приложения.

# РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ИНТЕРНЕТ-РЕСУРСА

# Разработка приложения

Было разработано приложение, основанное на ранее разработанной архитектуре.

Приложение состоит из двух основных папок: main, в которой хранится основной код проекта, и test, в которой хранятся тесты.

В папке main для серверной части главными папками являются:

* resources (в этой папке лежит файл application.properties с настройками проекта)
* java (в этой папке хранится пакет com.banksource.onlinebank с исходным кодом проекта)

В разрабатываемом приложении были реализованы следующие классы:

1. Были реализованы классы обработки запросов, поступающих с клиентской части – mainController и userController;
2. Были реализованы классы сервисы для работы с информацией о пользователях (UserService), картах (BankCardService), счетах пользователей (AccountService), транзакциях счетов (TransactionService), транзакциях карт (CardTransactionService), клиентах онлайн банка (ClientService).
3. Были реализованы классы работы с базой данных BankCardRepo, AccountRepo, UserRepo, TransactionRepo, CardTransactionRepo.
4. Был реализован класс логирования Aspect, использующий пакет Slf4j. Поведение прописано в logback.xml.
5. Был реализован класс обеспечения безопасности SecurityConfig, содержащий все необходимые настройки безопасности, предоставляемые пакетом com.banksource.onlinebank.security.

Основным классом, который содержит метод запуска проекта, является JavaApplication.

Полностью структура папок проекта представлена на рисунке 4.1

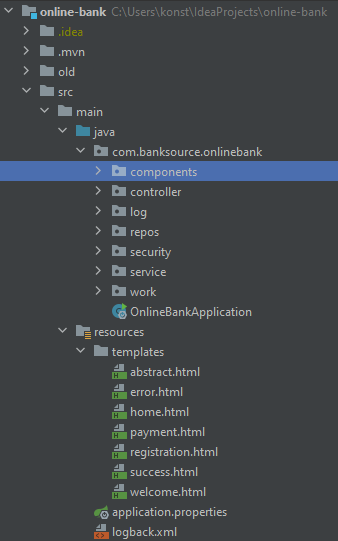


Рисунок 4.1.1 – структура приложения

# Тестирование приложения

Произведено пользовательское функциональное тестирование, согласно вариантам использования приведенных выше.

Запускаем приложение с помощью docker и переходим по адресу http://localhost:8000/welcome.

На рисунке 6.1 мы видим главную страницу приложения.

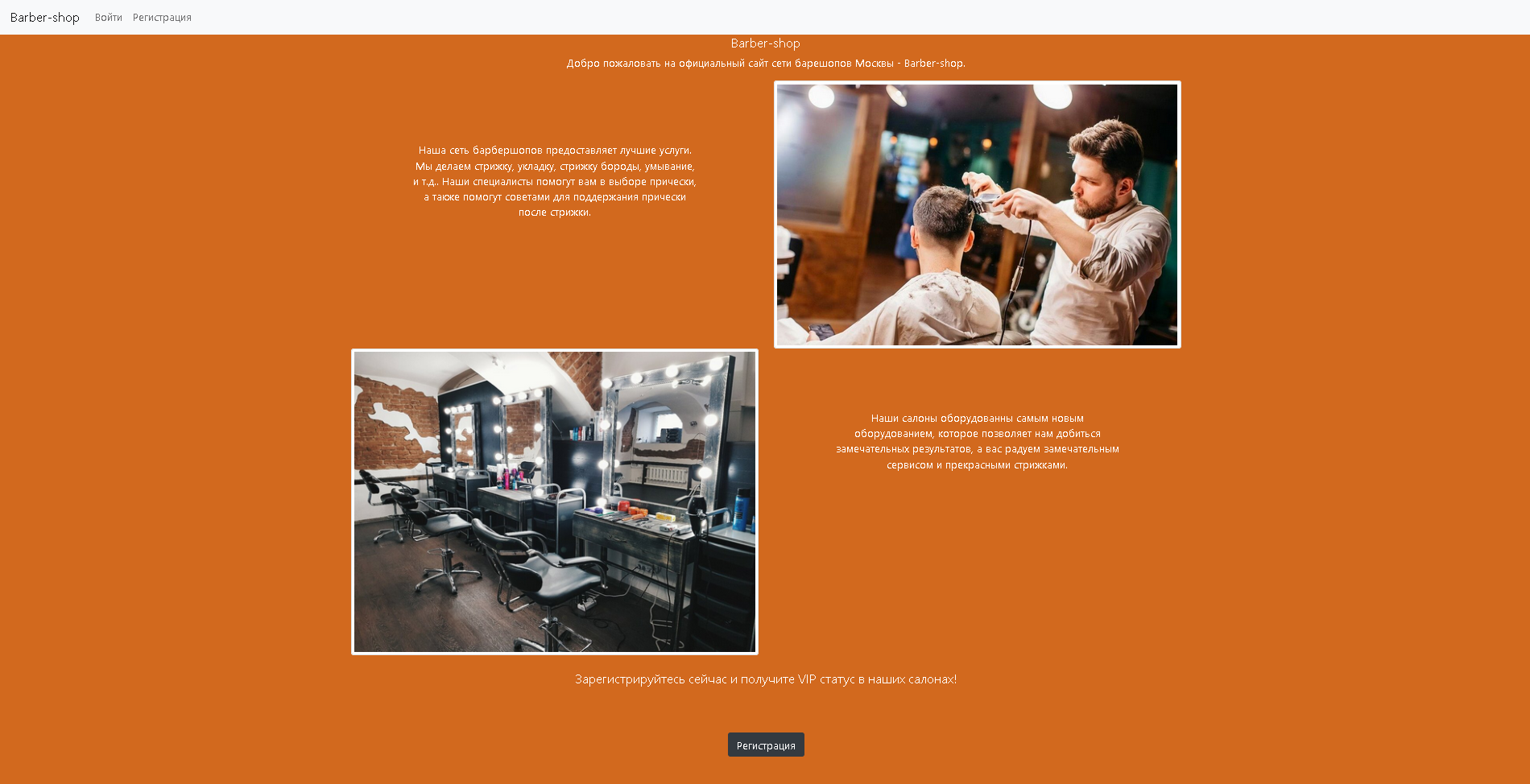


Рисунок 4.2.1 – главная страница

Далее для тестирования попытаемся зарегистрироваться в приложении. Для этого нажимаем кнопку регистрации. На рисунках 6.2 и 6.3 продемонстрированы окна регистрации и входа.

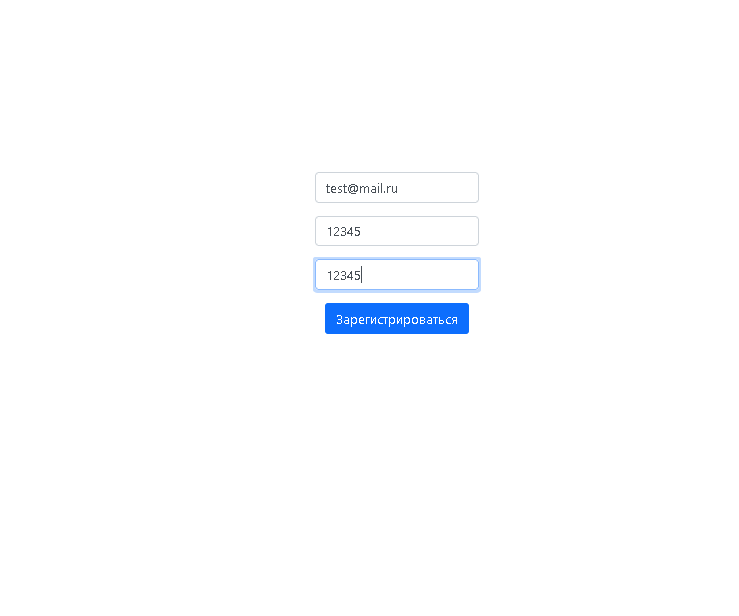


Рисунок 6.2 – страница регистрации

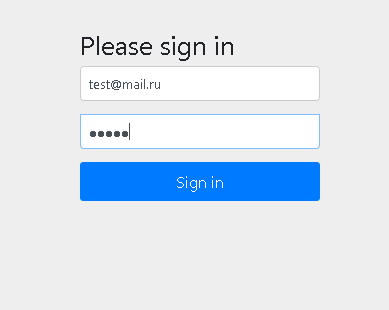


Рисунок 6.3 – страница входа

После входа мы попадаем на страницу записи на прием. На рисунке 6.4-6.5 продемонстрирован пользовательский интерфейс.

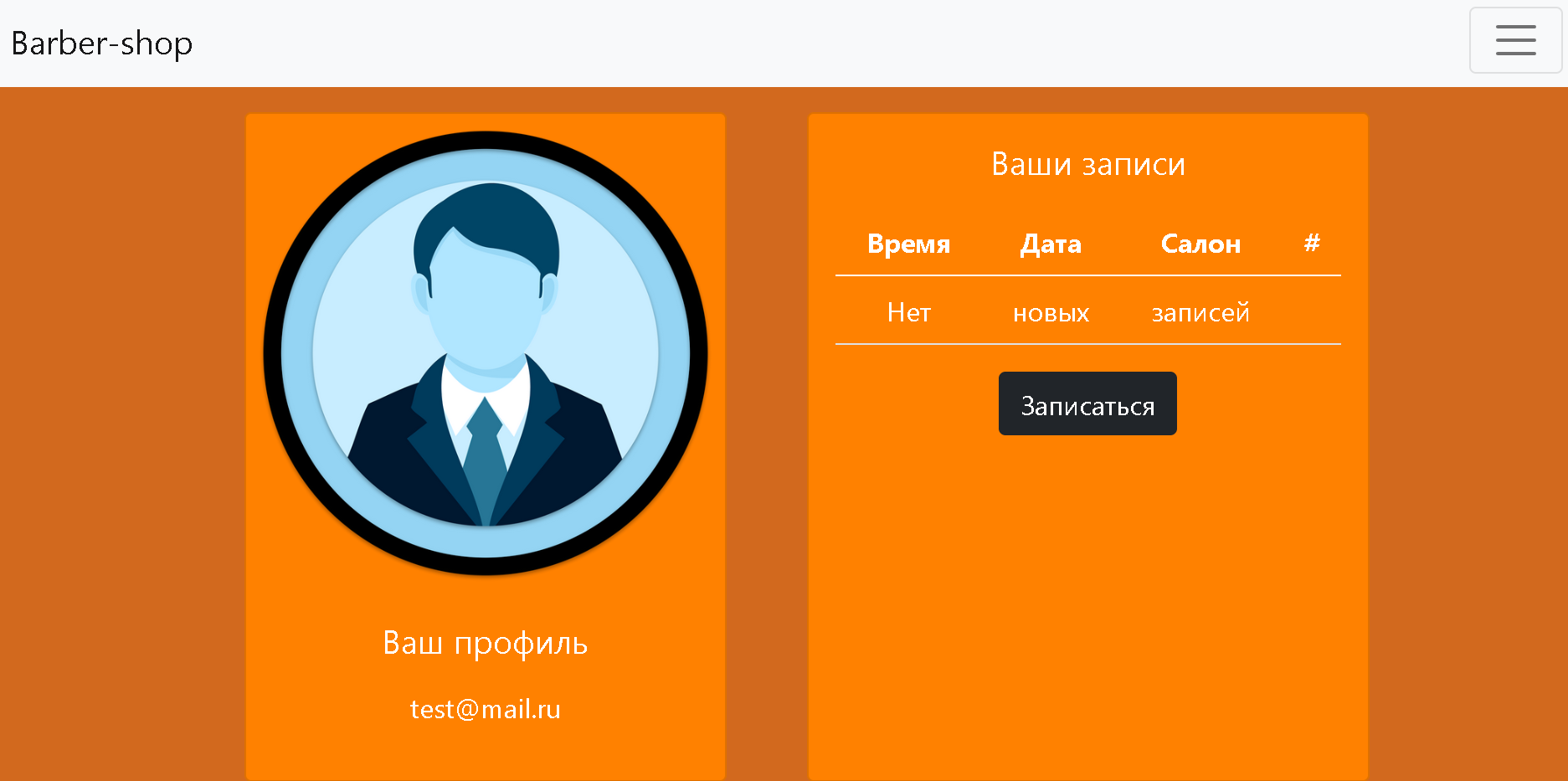


Рисунок 6.4 – пользовательский интерфейс

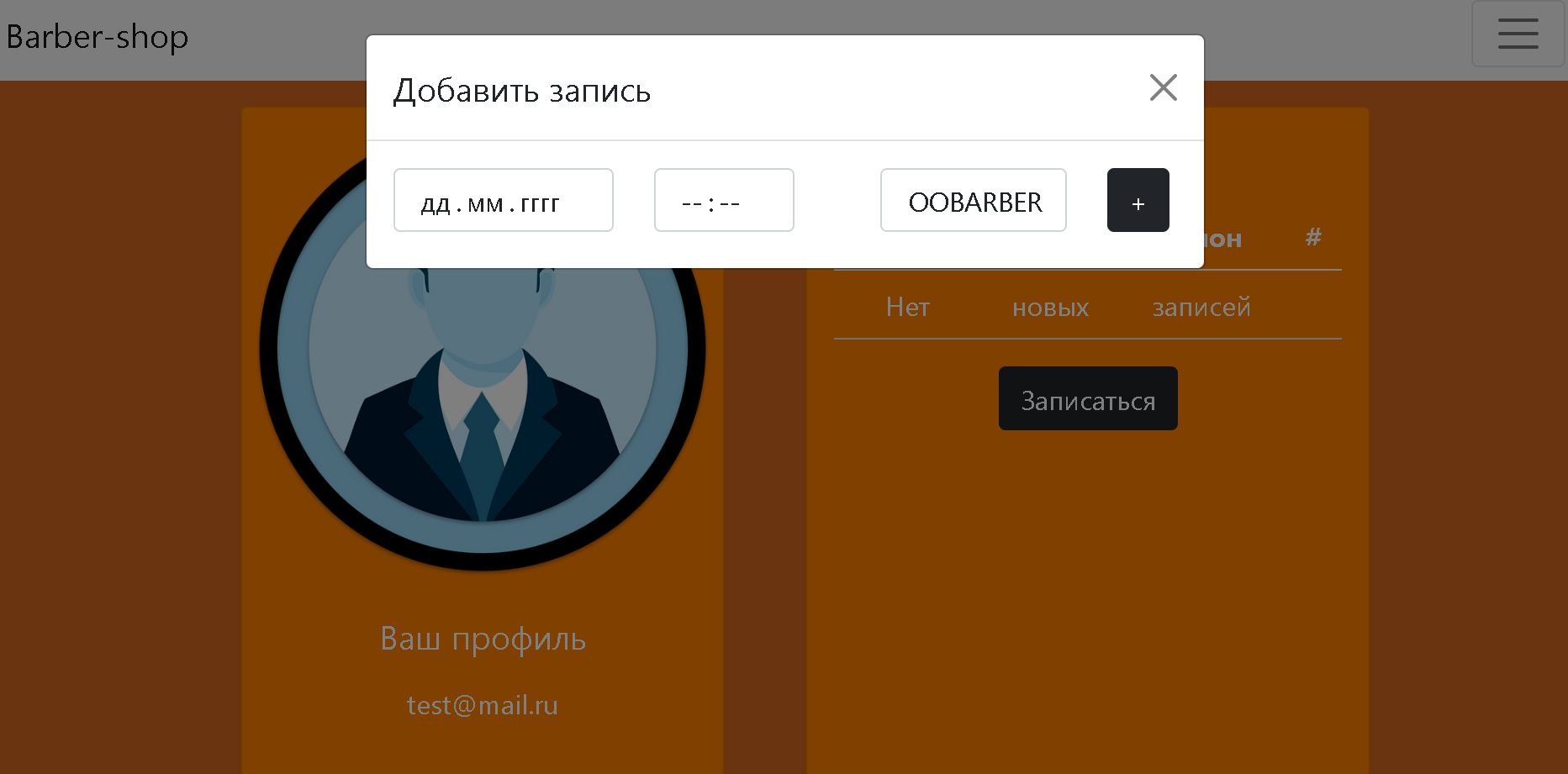


Рисунок 6.5 – пользовательский интерфейс

Теперь попытаемся записаться на прием в барбершоп. На рисунках 6.6 и 6.7 мы можем видеть на шу запись, из чего следует что сервис по записи пользователей работает корректно.

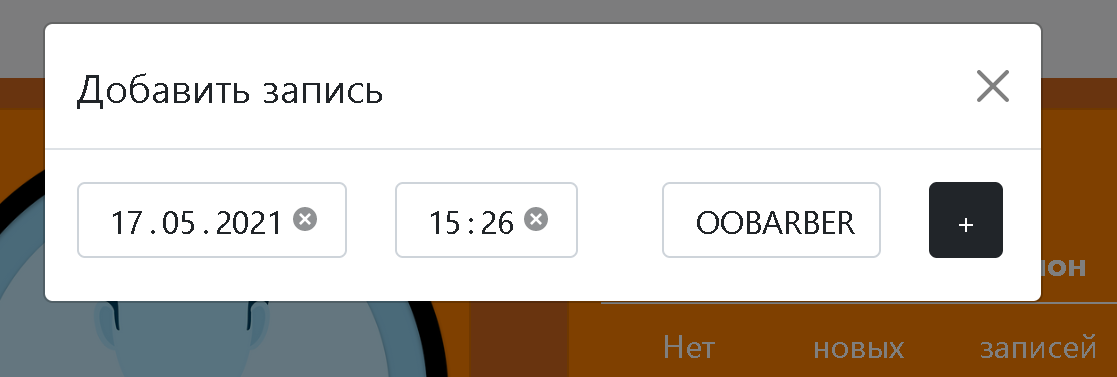


Рисунок 6.6 – попытка записи

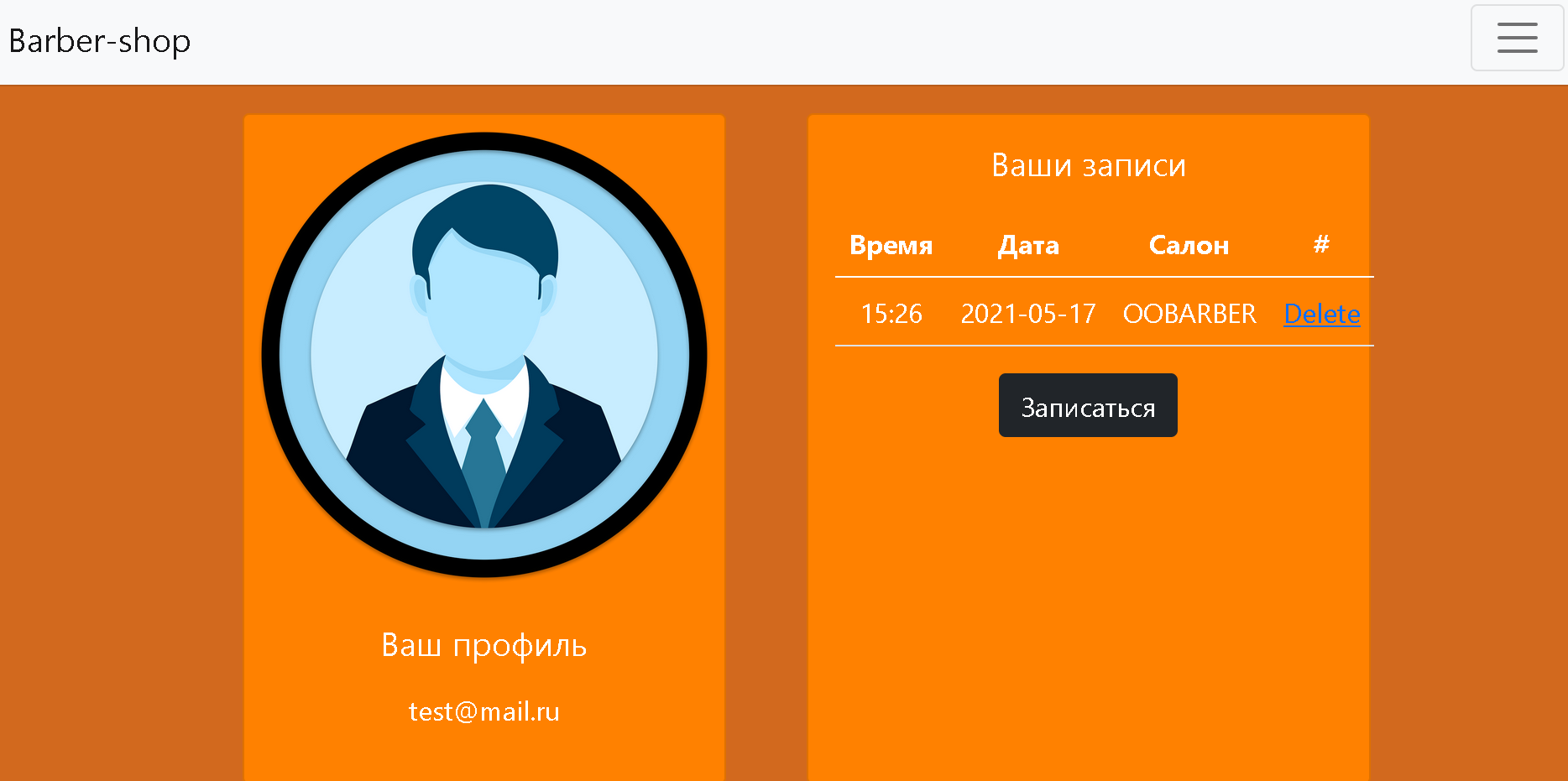


Рисунок 6.7 – пользовательский интерфейс с готовой записью

Также проверим работу суперпользователя и менеджера.

На рисунках 6.8 и 6.9 представлены интерфейсы суперпользователя и менеджера.

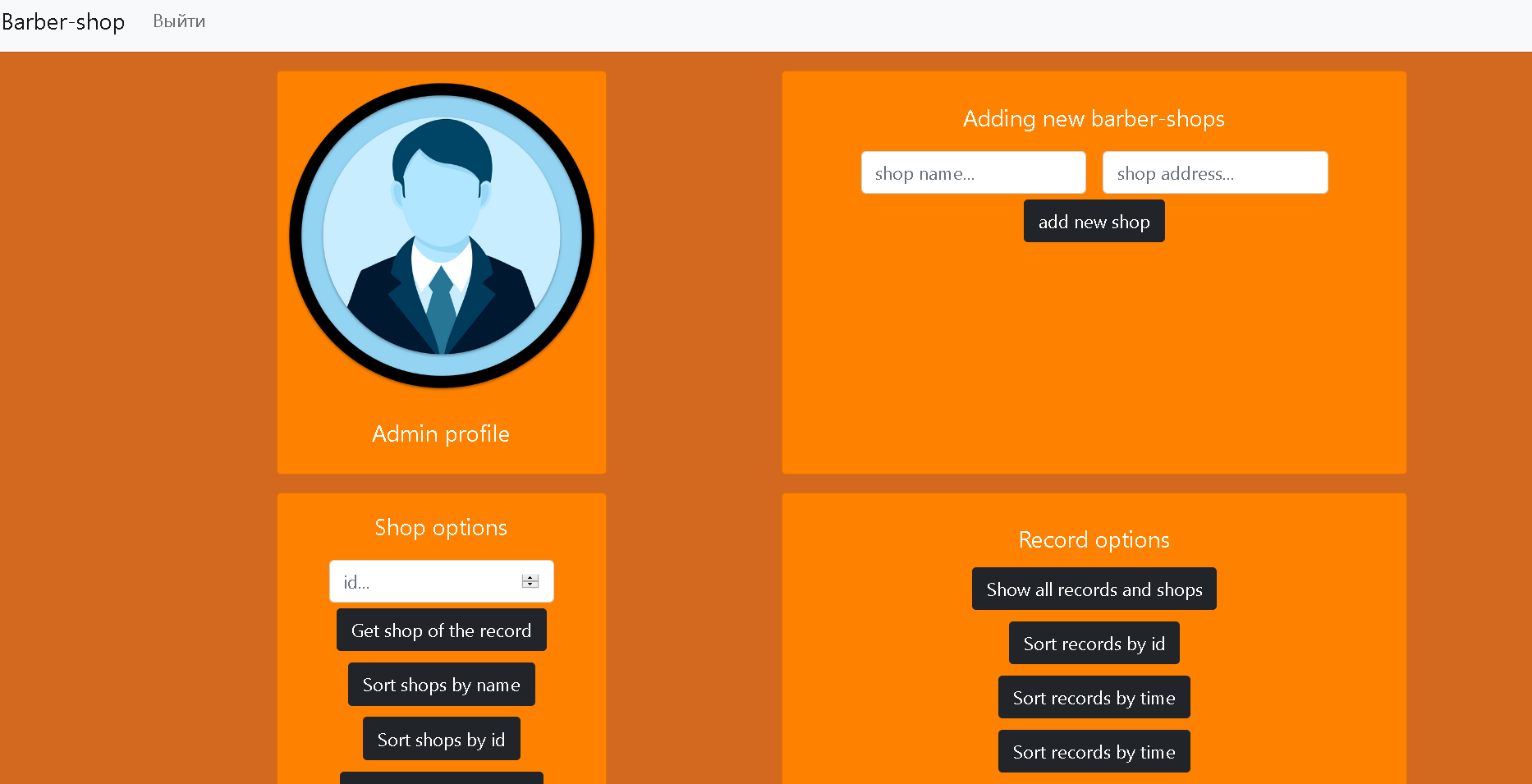


Рисунок 6.8 – интерфейс суперпользователя

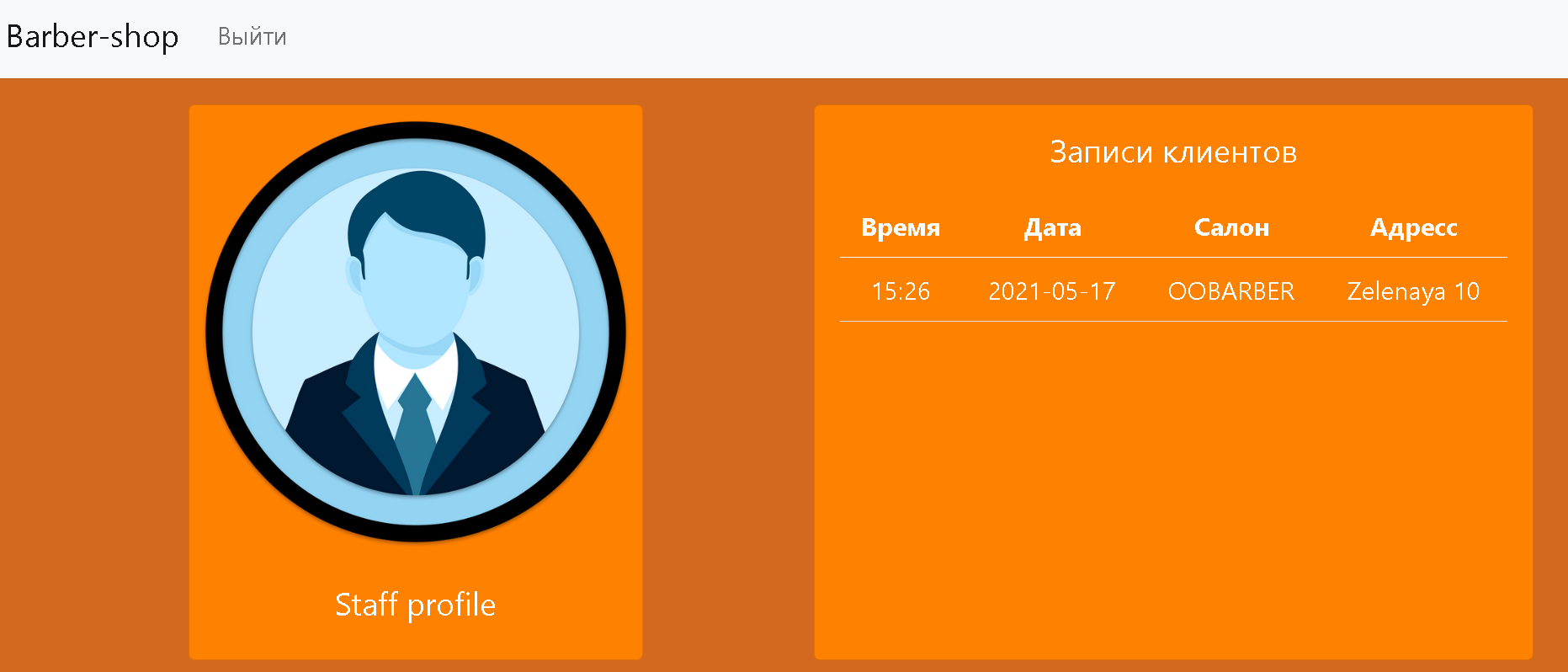


Рисунок 6.9 – интерфейс менеджера

# Выводы к разделу

На основе разработанной архитектуры, были созданы:

1. Компоненты приложения, включающие в себя основные сервисы приложения, реализующие бизнес-логику приложения;
2. Страницы, реализующие слой представления;
3. Контроллеры, реализующие логику взаимодействия представления и бизнес-логики.

В результате было получено приложение, отвечающее требованиям, полученным из главы - выбор и обоснование технологий.

В результате тестирования было уставлено, что приложение работает исправно без каких-либо ошибок и проблем.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения данной курсовой работы было проведено тестирование приложения «Барбершоп». Спроектированное приложение дает пользователям регистрироваться и записываться на прием в барбершоп.

Был проведён анализ предметной области с выявлением базовых функций для подобного приложения, а также определены требования, которые должны быть реализованы и протестированы в рамках системы заказчика.

Была разработана архитектура базы данных приложения в соответствии с предметной областью приложения.

Был установлен доступ и права для пользователей в соответствии с определенными ролями.

В ходе выполнения курсовой работы приложения было успешно контейнеризировано и протестировано.

В итоге было разработано приложение, которое дает пользователям регистрироваться и записываться на прием в барбершоп.

Исходные файлы приложения доступны в открытом репозитории на сервисе Github по адресу - https://github.com/DoshikKing/kursach\_Java\_Patterns.

Освоены компетенции:

ПК-1 - способен выполнять разработку и интеграцию программных

модулей и компонент, верификацию выпусков программных продуктов

информационных систем;

ПК-1.1 - Знать: методы и средства сборки модулей и компонент

программного обеспечения при создании информационных систем;

ПК-1.12 - Уметь: применять методы и средства создания программных

интерфейсов информационных систем;

ПК-1.14 - Владеть: разработкой процедур сборки модулей и компонент

программного обеспечения при внедрении информационных систем

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальный веб-сайт spring.io[Электронный ресурс]: Режим доступа: https://spring.io/ [Дата обращения: 16.05.21];
2. Статья Что такое Spring Framework? От внедрения зависимостей до Web MVC [Электронный ресурс]: Режим доступа: https://habr.com/ru/post/490586/ [Дата обращения: 14.05.21];
3. Документация Java Documentation [Электронный ресурс]: Режим доступа: https://docs.oracle.com/en/java/ [Дата обращения: 10.03.21];
4. Документация Postgres Documentation [Электронный ресурс]: Режим доступа: https://www.postgresql.org/docs/ [Дата обращения: 11.03.21];
5. Статья Изучение Spring Framework 100% через практику. Активные туториалы и язык разметки KML. Проект KciTasks (beta)[Электронный ресурс]: Режим доступа: https://habr.com/ru/post/347816/ [Дата обращения: 13.05.21];
6. Крейг Уоллс: Spring в действии [Книга]: Дата обращения: 17.05.21];
7. Markswebb [Электронный ресурс]: Режим доступа: https://markswebb.ru/report/internet-banking-rank-2020/ Дата обращения 16.11.21’

# ПРИЛОЖЕНИЕ

# Таблицы прецедентов

UC-01:

Сценарий

Регистрация нового пользователя

Код – UC-01

Уровень

Новый пользователь

Предусловие

Новый пользователь является клиентом банка (имеет открытые счета и/ или карты)

Основной сценарий

1. Клиент
   1. Обратился к оператору с заявлением о регистрации его в Интернет-банке (в заявлении НМТ клиента указан как логин)
2. Оператор
   1. Принял заявление
   2. Провел идентификацию клиента по документу, удостоверяющего личность (например, паспорт)
   3. В CBS клиенту включил услугу Интернет-банка
3. CBS
   1. Сохраняет признак активной услуги в БД
   2. Отправляет код для первичного входа в Интернет-банк (см. сценарий UC-02) клиенту на указанный в заявлении НМТ
   3. Отправляет уведомление в систему Интернет-банк о подключении услуги (НМТ и код подключения)
4. Интернет-банк
   1. Получает уведомление о подключении услуги
   2. Сохраняет в БД НМТ клиента, код регистрации (хешированный) и дату активации услуги
5. Оператор
   1. Сообщает клиенту об успешном подключении услуги
6. Клиент
   1. Получает код на свой НМТ
   2. Осуществляет первичный вход в Интернет-банк с использованием полученного кода (см. сценарий UC-02)

UC-02:

Сценарий

Первичный вход

Код – UC-02

Уровень

Новый пользователь

Предусловие

Успешный сценарий UC-01

Основной сценарий

1. Клиент
   1. Заходит на сайт Интернет-банка
   2. Выбирает функцию регистрации
   3. Вводит НМТ в качестве логина, задаёт новый постоянный пароль и его дубликат, код подтверждения, полученный в сценарии UC-01
   4. Нажимает кнопку регистрации
2. Интернет-банк
   1. Поучает НМТ, пароль и дубликат пароля, код подтверждения пользователя
   2. Проверяет наличие активной подключенной услуги для указанного НМТ и кода регистрации
   3. Сохраняет заданный постоянный пароль в БД
   4. В пользовательский интерфейс выдает сообщение об успешном завершении процедуры регистрации
   5. Перенаправляет на страницу входа (см. сценарий UC-03)
3. Клиент
   1. Получает доступ к своему Интернет-банку
   2. Использует функции Интернет-банка

UC-03:

Сценарий

Вход пользователя

Код – UC-03

Уровень

Активный пользователь

Предусловие

Успешный сценарий UC-02

Основной сценарий

1. Клиент
   1. Заходит на сайт Интернет-банка
   2. Выбирает функцию входа
   3. Вводит НМТ в качестве логина, пароль
   4. Нажимает кнопку входа
2. Интернет-банк
   1. Получает НМТ, пароль пользователя
   2. Проверяет наличие активной подключенной услуги для указанного НМТ
   3. Проверяет пароль пользователя
   4. Предоставляет доступ пользователю в систему

UC-04:

Сценарий

Получение списка счетов и карт

Код – UC-04

Уровень

Активный пользователь

Предусловие

Успешный сценарий UC-03

Основной сценарий

1. Клиент
   1. Клиент входит на главную страницу
2. Интернет-банк
   1. Запрашивает список активных счетов (номер счета и доступный остаток) из БД (см. сценарий UC-06)
   2. Запрашивает список активных карт (хешированные номер карты и доступный остаток) из БД (см. сценарий UC-06)
   3. Выводит на главной странице список активных карт и счетов пользователя

UC-05:

Сценарий

Перевод (C2A)

Код – UC-05

Уровень

Активный пользователь

Предусловие

Успешный сценарий UC-03, UC-04

Основной сценарий

1. Клиент
   1. Клиент выбирает операцию перевода
   2. Выбирает счет или карту для списания средств
   3. Выбирает счет или карту для зачисления средств
   4. Указывает сумму перевода
   5. Нажимает кнопку перевода
2. Интернет-банк
   1. Получает параметры перевода
   2. Рассчитывает комиссию согласно тарифам
   3. Проверяет не превышен ли суточный лимит (сумма по расходным транзакциям счета списания)
   4. Выводит общую сумму (сумма операции и рассчитанная комиссия) перевода пользователю
3. Клиент
   1. Подтверждает выполнение операции
4. Интернет-банк
   1. Проверяет не превышен ли суточный лимит (сумма по расходным транзакциям счета списания)
   2. Отправляет запрос в CBS для авторизации расходной операции по счету или карте списания
5. CBS
   1. Получает запрос, выполняет необходимые проверки, авторизует операцию
   2. Возвращает результат в Интернет-банк
   3. Выполняет сценарий UC-06
6. Интернет-банк
   1. Сообщает клиенту об успешно совершенной операции
   2. Выполняет сценарий UC-04

UC-06:

Сценарий

Получение списка счетов и карт из CBS

Код – UC-06

Уровень

Интернет-банк

Предусловие

Успешный сценарий UC-01

Основной сценарий

1. CBS
   1. Для всех клиентов с активной услугой Интернет-банк, выполняет нотификацию системы Интернет-банка об изменении статусов карт и счетов, доступных остатков карт и счетов, исполненных транзакций
2. Интернет-банк
   1. Получает нотификацию и сохраняет актуальную информацию в БД о картах, счетах и транзакциях

UC-07:

Сценарий

Получение выписки по счетам и картам

Код – UC-07

Уровень

Активный пользователь

Предусловие

Успешный сценарий UC-03, UC-04

Основной сценарий

1. Клиент
   1. Клиент запрашивает выписку по своим счетам и картам
2. Интернет-банк
   1. Запрашивает список транзакций активных счетов (номер счета и доступный остаток) из БД
   2. Запрашивает список транзакций активных карт из БД
   3. Выводит на главной странице список выписку по активным карам и счетам пользователя

UC-08:

Сценарий

Выход из Интернет-банка

Код – UC-08

Уровень

Активный пользователь

Предусловие

Успешный сценарий UC-03

Основной сценарий

1. Клиент
   1. Клиент нажимает кнопку выхода из Интернет-банка
2. Интернет-банк
   1. Выполняет процедуры выхода пользователя из системы
   2. Выводит сообщение об успешном выходе из системы пользователю