

ПРОГРАММНЫЙ МОДУЛЬ РАСЧЕТНЫХ ОПЕРАЦИЙ БАНКОВ РЕСПУБЛИКИ БЕЛАРУСЬ

Яцынович Е.А., студент

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Парамонов А.И. – канд. техн. наук, доцент, зав. каф. ИСиТ

В работе рассматривается проблема организации денежных переводов в новых условиях функционирования банковской системы. Предлагается решение в виде разработанного программного модуля расчетных операций, который предназначен для проведения денежных переводов с электронного кошелька на карты российских банков, а также перевод средств с зарубежных карт на электронный кошелек.

Денежные переводы – это уникальная банковская операция. Она очень удобна для тех, кто хочет отправить денежные средства своим близким, либо планирует совершить крупную покупку в другом городе или стране, но при этом не хочет перевозить большую сумму денег. Можно, конечно, перевести деньги почтовым переводом или же перевести их со счета на счет, но лучше воспользоваться современными банковскими услугами – системами денежных переводов.

С каждым годом отечественные системы переводов составляют все большую конкуренцию зарубежным аналогам. И если раньше мы знали, как моментально перевести деньги в другой город или страну, то теперь выбор систем переводов настолько велик, что трудно остановиться на каком-то одном варианте.

Предлагаемый программный модуль расчетных операций предоставляет функционал для проведения денежных переводов с электронного кошелька на карты российских банков и дополнительно есть возможность перевода средств с зарубежных карт на электронный кошелек.

Программный модуль является частью микросервиса осуществляющего проведение банковских операций. Архитектура системы имеет структуру, которая организует безопасность работы всей системы, а также позволяет расширять ее, для реализации поставленных задач. Все это обеспечивает использование архитектурного стиля микросервисов [1]. Общая схема архитектуры системы для обеспечения работы микросервиса представлена на рисунке 1.

Последовательность осуществления запроса на микросервис для перевода средств:

- Клиент посылает запрос с клиентского приложения на прокси-сервер посредством протокола https.
- Proxy сервер анализирует запрос и отправляет на сервер SECURE GATE.
- SECURE GATE снимает шифрование с запроса, зашифровывает его с помощью библиотек для криптографии и отправляет его на следующий прокси сервер.

- Прокси сервер пересылает запрос на менее загруженный микросервис (Microservice 1 и Microservice 2), при этом количество микросервисов зависит от нагрузки системы.
- Микросервис отправляет/получает данные с сервиса зарубежного банка.
- Микросервис взаимодействует с базой данных.
- Возвращает данные, выполняя проход по предыдущему пути (прокси / шифрование / передача данных клиенту).

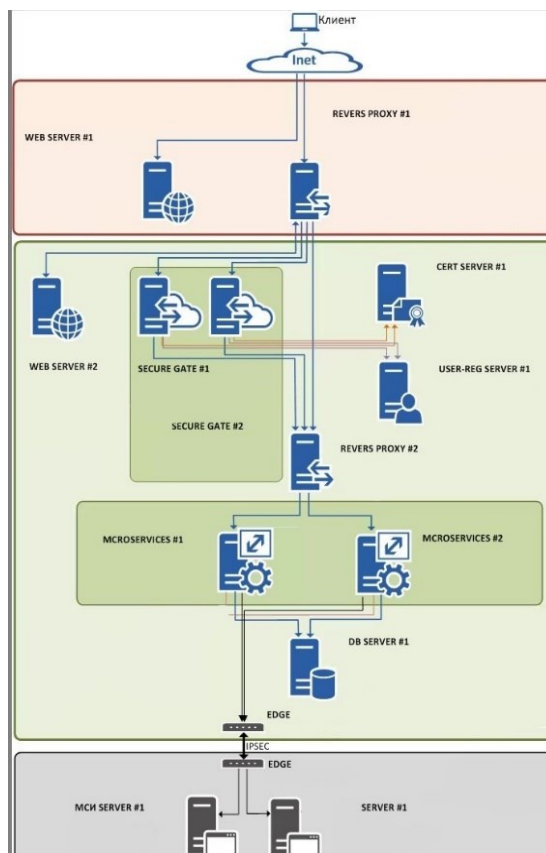


Рисунок 1 – Общая схема архитектура системы

Архитектура программного модуля использует шаблон проектирования модель представление-контроллер MVC. Данная архитектура является основой работы технологии JAVA SpringBoot [2]. Шаблон архитектуры MVC представляет собой приложение, разделенное на три части: модель, представление и контроллер. Представление отвечает за контент, отображаемый на экране пользователя, и текстовым пользовательским вводом. Контроллер непосредственно обрабатывает данные, которые были введены пользователем, и передает необходимые команды модели по изменению данных или представлению для изменения отображаемых элементов. Модель управляет данными программного средства, а также отвечает на инструкции о смене состояния [3].

Сервис по своей работе взаимодействует с большим количеством данных. Для хранения информации принято решение использовать реляционную СУБД ORACLE. Базы данных, с которой взаимодействует микросервис, представляет собой вертикальный тип.

Данные хранятся всего в несколько таблицах, для разделения данных в таблицах существуют специальные поля (типы), которые характеризуют каждую запись. В базе данных, с которой взаимодействует модуль, хранение информации организована с помощью главных и вспомогательных таблиц. В главных таблицах содержатся объекты, которые характеризуются элементами из вспомогательных таблиц (связь один ко многим). Объектами могут быть клиенты, платежи, операции, переводы, все эти объекты характеризуются специальным полем, определяющим тип записи, вспомогательные элементы организованы во вспомогательных таблицах аналогично.

Использование вертикальной архитектуры предусматривает довольно сложный доступ к данным, стандартные запросы выборки включают в себя большое количество условий, поэтому

таблицы индексируются. Выбор вертикальной архитектуры обусловлен необходимостью масштабирования базы данных [4].

Алгоритм осуществления перевода включает в себя расчет комиссии, представление клиенту суммы перевода включая комиссию, взаимодействие с сервисами банка получателя(отправителя), взаимодействие с базой данных. Комиссию за перевод устанавливает банк, она имеет нижнюю и верхнюю границы, определяющие сумму минимальной и максимальной комиссии.

Взаимодействие с сервисами банка получателя перевода заключается в передаче данных по переводу и получению его статуса. По статусу мы узнаем о состоянии операции и в зависимости от состояния пересылаем результат клиенту. Клиент наблюдает результат перевода в режиме текущего времени, но механизм перевода средств со счета на счет в банке происходит по расписанию.

Использование данного программного модуля расширяет функционал электронного кошелька, что повышает его привлекательность для использования клиентом, что соответственно влияет на экономическую выгоду банка.

Список использованных источников:

1. Chris Richardson, *Microservice Architecture* [Электронный ресурс] – Режим доступа: <https://microservices.io/> – Дата доступа: 26.03.2023
2. *Spring Boot по-русски* [Электронный ресурс] – Режим доступа: <http://spring-projects.ru/projects/spring-boot/>. – Дата доступа: 26.03.2023
3. Знакомство с паттерном MVC (Model-View-Controller) [Электронный ресурс] – Режим доступа: <https://javarush.com/groups/posts/2536-chastjh-7-znakomstvo-s-patternom-mvc-model-view-controller>– Дата доступа: 26.03.2023
4. Фрагментация базы данных: основные методы и случаи использования [Электронный ресурс] – Режим доступа: <https://www.8host.com/blog/fragmentaciya-bazy-dannyx-osnovnye-metody-i-sluchai-ispolzovaniya/>– Дата доступа: 26.03.2023