

ПРОГРАММНЫЙ МОДУЛЬ БАНКОВСКОЙ СИСТЕМЫ НА ОСНОВЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ

Белова А.В., студент

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

Сицко В.А. – маг. техн. наук, ст. препод. каф. ИСИТ

Данный проект представляет собой интегрированное технологическое решение, модуль для микросервисной архитектуры банковской системы. Цель модуля автоматизация ключевых процессов взаимодействия с пользователями и продуктами, обеспечение безопасности, производительности и масштабируемости. Результатом работы является полноценный интегрированный модуль банковской системы, соответствующий стандартам безопасности, законодательным нормам, обеспечивающий эффективное взаимодействие в рамках банковской системы.

Банковская система представляет собой сложную структуру элементов, многофункциональное образование финансовых институтов, включающую в себя банки, техническую базу, обеспечивающую передачу и обработку финансовых операций, платежи, кредитование, управление счетами, подразумевает под собой функционирование банков и кредитных учреждений как единого денежно-кредитного механизма. В современном мире банковские системы играют ключевую роль в экономике государства, обеспечивая финансовую устойчивость и эффективное функционирование экономики.

Современный уровень развития финансового рынка предопределяет необходимость разработки и внедрения новых подходов к системе формирования и обращения различных видов ресурсов, которыми оперирует кредитная организация – денежных, материальных, трудовых, интеллектуальных и пр. [1]

Проект представляет собой комплексное банковское решение, включающее в себя оформление и работу с вкладами, счетами и карточками клиентов, получение курсов валют, списков банковских продуктов, реализацию безопасной системы авторизации с возможностью удаленной регистрации.

Проект направлен на интеграцию современных концепций микросервисной архитектуры для обеспечения гибкости и устойчивости системы, повышения качества обслуживания клиентов и эффективности банковских операций.

Микросервисная архитектура выбрана для банковского модуля из-за своей гибкости и масштабируемости. Каждый микросервис действует автономно, обеспечивая независимость и устойчивость к отказам.

Микросервисы широко используются для создания сложных приложений с развитой многофункциональностью путем объединения всех частей и их послышной компоновки в единое целое. Многие из нас могут не знать, что микросервисы — это подход к созданию единого приложения в виде набора небольших сервисов, где каждый сервис работает сам по себе (автономный процесс), но которые коммуницируют между собой как единое целое через API. [2]

Программный модуль предназначен для внедрения в микросервисную архитектуру банковской системы, представляет собой ключевой компонент, комплексный банковский модуль, предназначенный для оформления различных банковских продуктов счетов, карточек, вкладом.

Самым оптимальным сочетанием фреймворка и языка программирования для создания модуля на основе микросервисной архитектуры служит Java и Spring Framework, представляющий комплексный набор инструментов для построения приложений. Обеспечивает гибкую архитектуру, легкое создание, развертывание, масштабирование и управление микросервисами, безопасность.

Использование Java обеспечивает независимость от платформы, что позволяет запускать программы на различных устройствах. Обширная экосистема, состоящая из множества библиотек для разработки, позволяет находить необходимые решения.

Spring Framework обширный фреймворк для разработки на Java. Позволяет внедрять и управлять зависимостями, обеспечивает безопасность многопоточных приложений.

При разработке архитектуры программного модуля было принято решение использования шаблона архитектуры MVC. Данная архитектура является основой работы технологии JAVA SpringBoot

MVC – наиболее популярный и мощный шаблон архитектуры для разработки пользовательских интерфейсов. 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 модуля, а также отвечает на

инструкции о смене состояния (как правило, передаваемые контроллером). Модель выступает в качестве связующего звена между контроллером и представлением. [3]

Диаграмма развертывания – это тип UML-диаграммы, которая показывает архитектуру исполнения системы, включая такие узлы, как аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их. [4]

Диаграмма развертывания программного модуля представлена на рисунке 1.

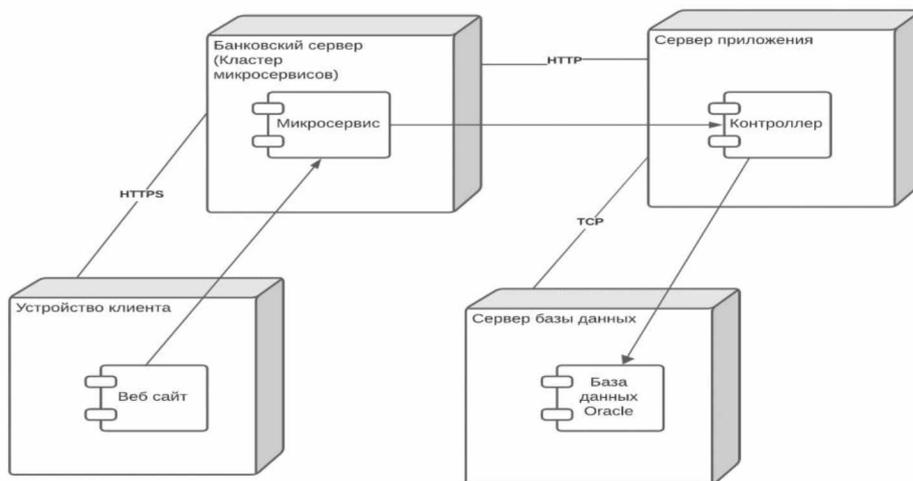


Рисунок 1 – Диаграмма развертывания программного модуля

Функциональным назначением проекта является возможность разрабатываемого модуля обеспечивать регистрацию и аутентификацию пользователей, оформление, управление и работа с банковскими продуктами такими как банковские счета, вклады, карточки, управление финансовыми операциями, администрирование и взаимодействие с другими сервисами микросервисного кластера.

Интерфейс пользователя представлен на рисунке 2.

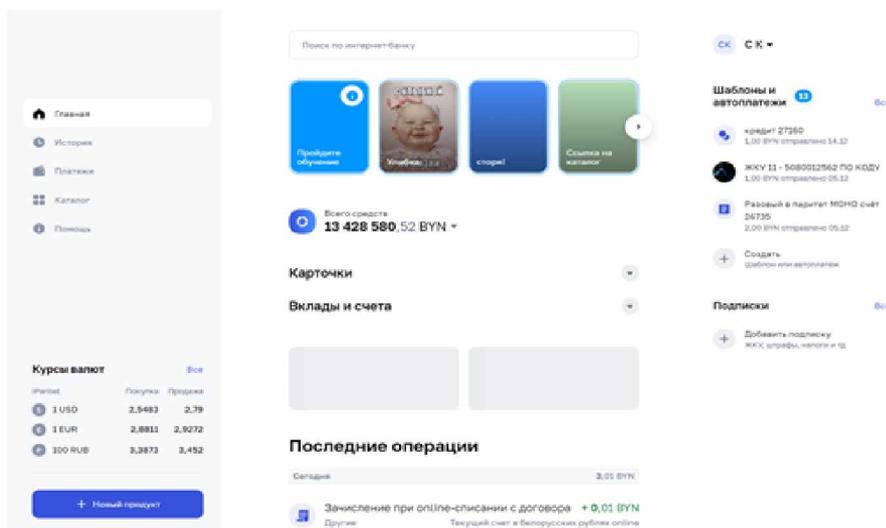


Рисунок 2 – Личный кабинет пользователя

В результате работы получен комплексный интегрированный банковский модуль, позволяющий банковской системе обеспечивать эффективное управление пользователями и финансовыми операциями, взаимодействие с другими межбанковскими и внешними сервисами.

Список использованных источников:

1. *Банковское дело и банковские операции: учебник для обучающихся в высших учебных заведениях по направлениям подготовки «Экономика и управление» / М. С. Марамыгин [и др] – Министерство науки и высшего образования Российской Федерации, Уральский государственный экономический университет. – Екатеринбург: Изд-во Урал. ун-та, 2021. – 567 с.*
2. *5 лучших Java Frameworks для микросервисов в 2022 году [Электронный ресурс] – Режим доступа: <https://highload.today/java-frameworks-microservice/>*
3. *Что такое архитектура MVC [Электронный ресурс] – Режим доступа: <https://wiki.merionet.ru/servermye-resheniya/100/chto-takoe-arhitektura-mvc-model-view-controller/>*
4. *А.П. Пашкевич, О.А. Чумаков/ Современные технологии программирования – Минск - 2007.*