

## СОЗДАНИЕ МИКРОСЕРВИСА «АЙЫЛ БАНК» ДЛЯ ИНТЕГРАЦИИ С ЭЛЕКТРОННЫМ КОШЕЛЬКОМ «ЭЛСОМ»

**Введение.** В современном мире использование электронных кошельков является актуальным и имеет множество преимуществ:

1. Электронные кошельки позволяют быстро и удобно проводить различные финансовые операции, такие как оплата покупок, переводы средств и оплата счетов.

2. Электронные кошельки обеспечивают высокий уровень безопасности, так как часто используются двухфакторная аутентификация и шифрование данных.

3. С помощью мобильных приложений для электронных кошельков, пользователи могут осуществлять платежи практически в любом месте, где есть доступ к интернету.

4. Многие электронные кошельки предоставляют инструменты для учета расходов и анализа финансов, что помогает пользователям более эффективно управлять своими финансами.

5. Возможность интеграции с другими сервисами, такими как магазины, системы лояльности и приложения.

6. Использование электронных кошельков устраняет необходимость в поиске банкомата или обменного пункта, что экономит время.

7. Электронные кошельки способствуют сокращению использования бумажных денег и чеков, что положительно влияет на экологию.

8. Многие электронные кошельки поддерживают международные операции и валюты, что упрощает мировые финансовые транзакции [1].

«Элсом» — это приложение, которое предоставляет пользователям возможность совершать платежи, переводы, проверять баланс и получать информацию о своих финансовых операциях.

Целью данного исследования является создание микросервиса «Айыл Банк» для интеграции с электронным кошельком «Элсом». Объектом исследования является микросерверная архитектура разработки программного обеспечения. Предметом исследования выступает процесс осуществления платежей и переводов, связанных с банком «Айыл Банк».

**Основная часть.** При создании микросервиса для интеграции с электронным кошельком «Элсом» необходимо решить следующие задачи:

1. Сделать доступ к электронному кошельку более удобным и интуитивно понятным для пользователей, путем предоставления удобного API и интерфейса.

2. Расширить функциональность электронного кошелька, предоставляя дополнительные сервисы, такие как управление финансами, статистика операций и др.

3. Осуществить интеграцию электронного кошелька с другими сервисами или платформами.

4. Обеспечить высокий уровень безопасности и конфиденциальности данных пользователей, предотвращая несанкционированный доступ и мошенничество.

5. Гарантировать соблюдение всех необходимых стандартов и законодательных требований, связанных с финансовыми операциями.

6. Предоставить качественную поддержку пользователям и оперативно реагировать на их запросы и вопросы.

Для создания микросервиса в EIP фреймворке будут использованы следующие средства разработки:

– Rabbit MQ — программный продукт, который передает сообщения между отправителем и получателем, т. е. служит посредником между сервисами;

– Redis — это нереляционная резидентная СУБД, хранящая данные в виде пар «ключ-значение»;

– Postman — это приложение, способное выполнять запросы API к любому API HTTP. Используется для тестирования и изучения API - интерфейсов [2];

– Java — строго типизированный объектно-ориентированный язык программирования общего назначения.

Для приложения выделены два варианта пользователей:

External Users — это пользователи, которые находятся вне вашей организации или системы и взаимодействуют с вашими ресурсами извне.

Internal Users — это пользователи, работающие внутри организации или системы, и имеющие доступ к внутренним ресурсам и данным.

Диаграммы вариантов использования приложения для External Users и Internal Users отображены на рисунках 1 и 2 соответственно.

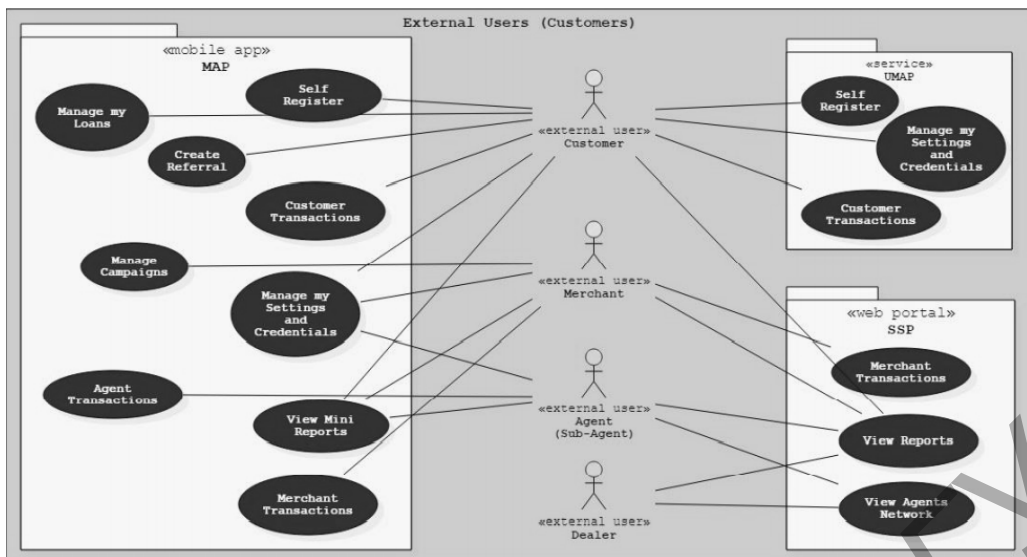


Рисунок 1 — Диаграмма вариантов использования для External Users

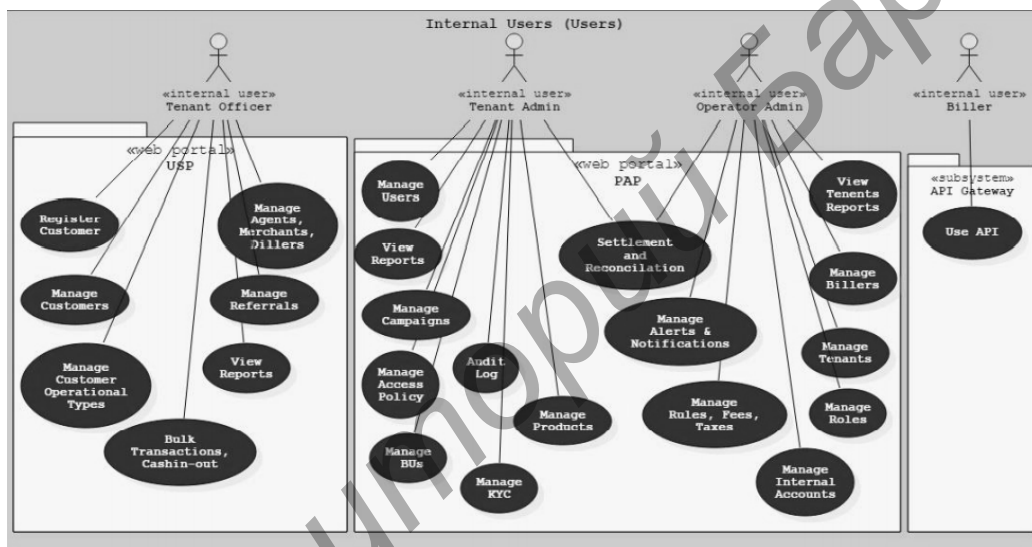


Рисунок 2 — Диаграмма вариантов использования для Internal Users

Для идентификации пользователя применяется Bearer Token. Bearer Token — это тип токена доступа, используемого в процессах аутентификации и авторизации, особенно в контексте веб-сервисов и API. Это безопасный токен, включаемый в заголовки HTTP-запросов для подтверждения личности вызывающего.

Для запросов, используются такие методы, как POST — для проверки получателя и GET — для проведения платежа. Система электронного кошелька предоставляет шлюз для проведения платежей от различных платежных терминалов, других систем партнеров и реализована в стиле RESTFULL API по протоколу HTTPS. Для проведения платежей партнер отправляет JSON объект, POST запросом в систему электронного кошелька и получает ответ в JSON формате. Для дополнительной аутентификации систему партнера используется пароль, выданный поставщиком услуг.

Пополнение кошелька состоит из двух шагов: валидация и пополнение. При валидации партнер отправляет первоначальные данные поставщику, где поставщик возвращает доступность клиента и нижний, верхний лимит на сумму пополнения. Данные лимиты терминал должен отобразить клиенту. После внесения суммы клиентом, партнер отправляет сумму и другую информацию для пополнения электронного кошелька

В проекте реализована аутентификация через механизм JSON Web Tokens. Пользователям выдаются токены после успешной аутентификации, которые подтверждают их идентичность и роли. Использованы аннотации Spring Security для ограничения доступа к определенным эндпоинтам в зависимости от роли пользователя.

Весь трафик между клиентом и сервером защищен протоколом HTTPS с использованием SSL-сертификата. Это обеспечивает шифрование данных, передаваемых между сторонами.

Для запросов к базе данных MS SQL используется NamedParameterJdbcTemplate из Spring JDBC, что обеспечивает параметризованные запросы и предотвращает SQL-инъекции. Введено регулярное резервное

копирование базы данных MS SQL. Резервные копии хранятся на отдельном сервере с ограниченным доступом.

Реализована кастомизированная обработка ошибок через глобальные обработчики исключений Spring. Это позволяет предоставлять клиентам информацию об ошибках без раскрытия чувствительных данных.

**Заключение.** Разработанный сервис «Айыл банк» для интеграции с электронным кошельком «Элсом»:

- обеспечивает удобство и быстроту платежей, так как пользователи могут оплачивать счета, делать покупки и переводы с минимальными усилиями;
- обеспечивает повышенную защиту данных и финансовых операций при онлайн-транзакциях;
- обеспечивает удобную интеграцию мобильного кошелька с магазинами и сервисами;
- обеспечивает интеграцию не только с банковскими системами, но и с другими сервисами, такими как мобильные приложения, магазины, системы лояльности и другие;
- позволяют ускорить процесс оплаты и избежать неудобств, связанных с использованием наличных денег.

#### Список цитируемых источников

1. Ревенков, П. В. Финансовый мониторинг: управление рисками отмывания денег в банках / П. В. Ревенков, А. Б. Дудка, А. Н. Воронин, М. В. Каратаев. — М. : КНОРУС, ЦИПСИР, 2021. — 335 с.
2. Hoffman, K. Building Microservices with ASP.NET Core / Kevin Hoffman. — O'Reilly Media, 2017. — 232 с.

УДК 510.64:624.012.3

**П. Ю. Тихончук**

*Учреждение образования «Брестский государственный технический университет», Брест, Республика Беларусь*

*Научный руководитель  
Ю. С. Дордюк*

### АПРОБИРОВАНИЕ ТЕОРИИ НЕЧЕТКИХ МНОЖЕСТВ ПРИ ОБСЛЕДОВАНИИ РЕАЛЬНЫХ СТРОИТЕЛЬНЫХ КОНСТРУКЦИЙ

**Введение.** В настоящее время одной из наиболее актуальных градостроительных проблем является качество строительства, определяющее срок службы зданий. Присвоение строительной конструкции той или иной категории при визуальном осмотре по выявленным дефектам и повреждениям носит достаточно субъективный характер и требует для обследования огромного опыта у эксперта.

**Основная часть.** Один из современных подходов, используемых в различных задачах принятия решений в условиях неопределенности, основан на применении инструментария теории нечетких множеств Л. А. Заде [1, с. 15]. Для проведения научных исследований была использована система нечеткого вывода в рамках среды MatLab со встроенным пакетом Fuzzy Logic Toolbox. [2, с. 534]

В качестве входных параметров системы нечеткого вывода рассматривались 14 нечетких лингвистических переменных: «Распространение нормальных/наклонных трещин», «Положение нормальных/наклонных трещин», «Распространение продольных коррозионных трещин», «Наличие коррозионных повреждений», «Поверхностные повреждения бетона» и «Распространение продольных силовых трещин в бетоне сжатой зоны элемента», «Отношение толщины защитного слоя бетона к диаметру арматурного стержня», «Образование трещин нормального отрыва», «Образование продольных коррозионных трещин», «Степень коррозионного повреждения», «Прогибы, перемещения», «Визуальная инспекция», «Базовые испытания», «Наличие проектной документации» а в качестве выходных параметров — нечеткие лингвистические переменные «Уровень повреждения конструкции», «Класс повреждения конструкции».

Для каждого термина были определены типы функций принадлежности, таким образом, чтобы при пересечении двух функций они пересекались в точке 0,5 по оси ординат, но и соответствовали границам по оси абсцисс. После задания 98 правил нечеткого вывода выдавался результат нечеткого вывода для конкретных значений входных переменных.

Полученные результаты зависимостей между основными базисными переменными и соответствующие диапазоны их граничных значений легли в основу разработки диагностической карты для ввода исходных данных. Данная диагностическая карта состоит из двух основных этапов «Визуальная инспекция» и «Базовые испытания», по результатам оценивания которых определяется уровень повреждения конструкции. Полученные уровни повреждения конструкции с учетом наличия или отсутствия проектной документации определяют класс повреждения конструкции.

На основании описанной методики разработана диагностическая карта, которая состоит из двух основных этапов «Визуальная инспекция» и «Базовые испытания», по результатам оценивания которых определяется уровень повреждения конструкции. Полученные уровни повреждения конструкции с учетом наличия или отсутствия проектной документации определяют класс повреждения конструкции.