

СЕКЦИЯ «ПРОБЛЕМЫ ЭКОНОМИЧЕСКОЙ ИНФОРМАТИКИ»

УДК 004.04

МУЛЬТИБАНКОВСКИЕ ПЛАТЕЖНЫЕ ПЛАТФОРМЫ И ПРОГРАММНАЯ ПОДДЕРЖКА МОНИТОРИНГА И КОНТРОЛЯ ДВИЖЕНИЯ ДЕНЕЖНЫХ СРЕДСТВ В РЕАЛЬНОМ ВРЕМЕНИ

Артишевский А.Д.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сторожев Д.А. – старший преподаватель

Аннотация. Казначейский агрегатор – набор решений, обеспечивающих упорядоченное подключение по всем корпоративным банковским активностям, включая отчеты о балансе и банковские переводы, через сеть SWIFT, другие сети и прямые подключения. Подключения, предоставляемые казначейскими агрегаторами, значительно упрощают процесс управления несколькими банковскими порталами или платежными системами. Казначейские агрегаторы управляют процессом платежей с момента их возникновения до доставки и составления отчетов. Большинство агрегаторов также предлагают возможность управления денежными средствами. Разработанное программное средство ориентировано на сотрудников банков, которые смогут анализировать и проводить мониторинг движения денежных средств, ориентируясь на данные программы.

Ключевые слова. Банки, агрегатор, SWIFT, транзакции, денежные средства, нотация IDEF0, диаграммы UML, база данных, фрэймворк, веб-приложение.

Агрегаторы обеспечивают централизацию, оптимизацию и упрощение процесса подключения банков. Компании необходимо только одно соединение с казначейским агрегатором, который будет управлять поддержкой и обслуживанием всех подключений с банками. Казначейский агрегатор – наиболее привлекательное решение для компаний, ведущих деятельность со множеством банков в различных географических регионах и сталкивающихся с проблемами разнообразия форматов данных. Наиболее популярные мультибанковские платежные платформы и агрегаторы: НРД Транзит 2.0, CyberFT, Банк России (СПФС), Универсальная платежная система корпораций (УПКС), IMS Payments (ECS Fin), Fides EFT, SWIFT.

На рисунке 1 показана схема функционирования мультибанковской платежной платформы.



Рисунок 1 – Схема функционирования мультибанковской платежной платформы

Важным направлением в развитии взаимодействия банков с корпоративными клиентами в сфере осуществления расчетов является использование инфраструктуры SWIFT, которая помимо традиционных преимуществ SWIFT (надежности, скорости, безопасности и финансовой ответственности) позволяет крупным корпорациям применять принцип "единого окна" при работе с несколькими банками.

Один терминал SWIFT, установленный у клиента, позволяет обмениваться внутренними и международными сообщениями со всеми обслуживающими банками.

В связи с усиленными мерами безопасности и увеличением давления со стороны регулирующих органов на рынке корпорации стремятся уменьшить затраты и эксплуатационный риск, централизовать и автоматизировать операции, чтобы повысить эффективность и улучшить общий контроль.

SWIFT предоставляет корпорациям платформу для обмена финансовой информацией (платежи, расчеты по ценным бумагам, отчетность) с банками через единую безопасную стандартизированную платформу вместо многократных соединений.

Преимущества использования инфраструктуры SWIFT для корпораций:

повышенная автоматизация и сквозная обработка данных (STP);

контроль/безопасность/надежность;

соответствие нормам по противодействию отмыванию денег и финансированию терроризма.

При этом используются стандарты финансовых сообщений SWIFT, широко распространенные в банковской индустрии и охватывающие все сегменты финансового рынка, начиная с клиентских платежей и отчетов по счетам и заканчивая подтверждениями форексных и депозитных сделок [1].

Описание системы с помощью IDEF0 называется функциональной моделью. Функциональная модель предназначена для описания существующих бизнес-процессов, в котором используются как естественный, так и графический языки. Для передачи информации о конкретной системе источником графического языка является сама методология IDEF0.

Контекстная диаграмма верхнего уровня данного программного продукта представлена на рисунке 2.

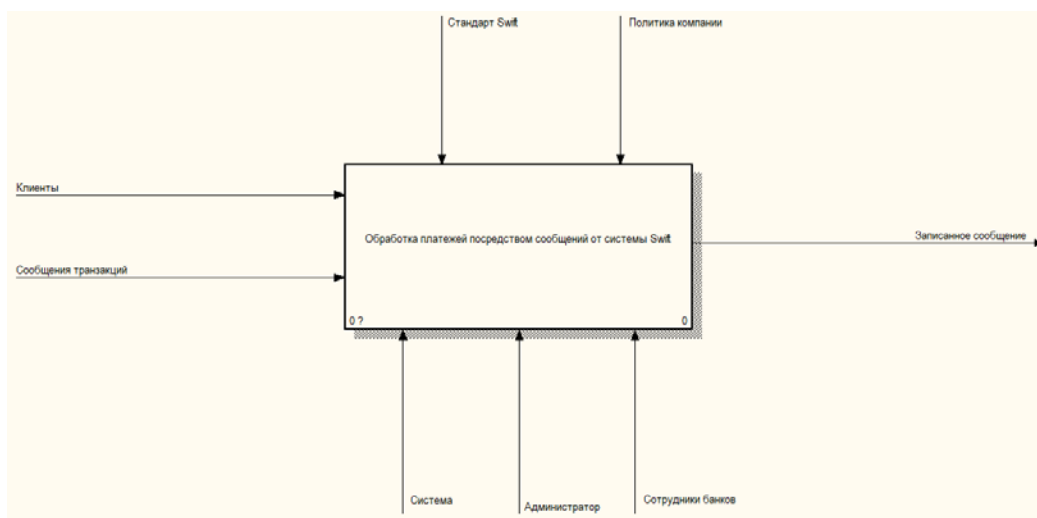


Рисунок 2 – Контекстная диаграмма верхнего уровня «Обработка платежей посредством сообщений от системы Swift»

Входными потоками данного уровня являются «Клиенты» и «Сообщения транзакций». В качестве выходного потока выступает «Записанное сообщение». В данной диаграмме присутствует 2 вида управления: «Стандарт Swift» и «Политика компании». Механизмом являются «Сотрудники банков», «Администратор» и «Система».

Диаграмма верхнего уровня декомпозируется на 5 блоков «Отправить сообщение», «Извлечь данные», «Найти соответствующий аккаунт», «Создать запись о платеже» и «Вернуть ответ на запрос» (см. рисунок 3).

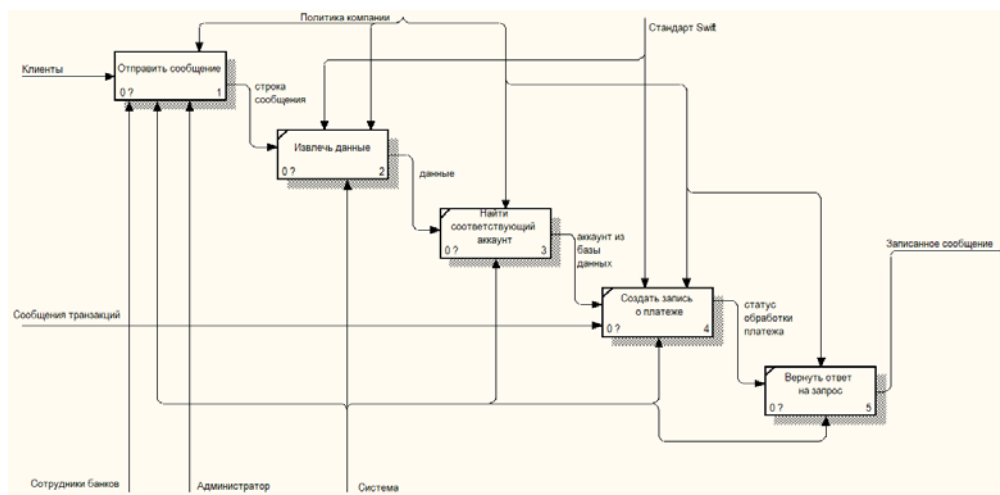


Рисунок 3 – Контекстная диаграмма верхнего уровня «Обработка платежей посредством сообщений от системы Swift»

Сотруднику и администратору необходимо отправить сообщение. Затем следует извлечь данные и найти соответствующий аккаунт. Теперь мы получим нужный аккаунт из базы данных и можем создать запись о платеже и вернуть ответ на запрос ожидающему клиенту.

На диаграмме вариантов использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых прецедентов. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой [2].

Диаграмма вариантов использования программного продукта представлена на рисунке 4.

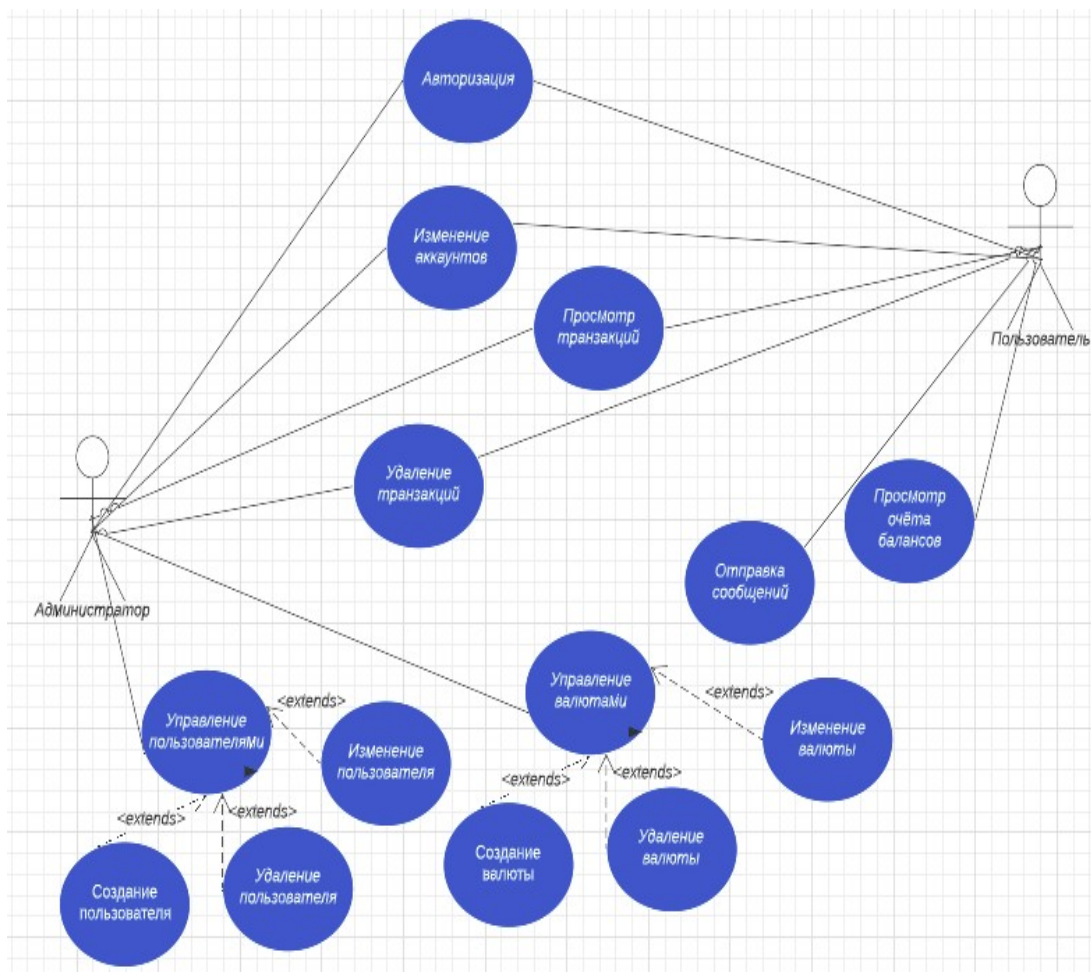


Рисунок 4 – Диаграмма вариантов использования

Серверная часть приложения реализована на Python. У языка Python есть много преимуществ перед другими языками программирования, что позволяет решать с его помощью практически любые задачи.

Для написания приложения использовалась среда разработки PyCharm. PyCharm – интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA. PyCharm – это кроссплатформенная среда разработки, которая совместима с Windows, MacOS, Linux.

В результате проделанной работы был создан удобный сайт для мониторинга движения денежных средств. Главная страница сайта представлена на рисунке 5.

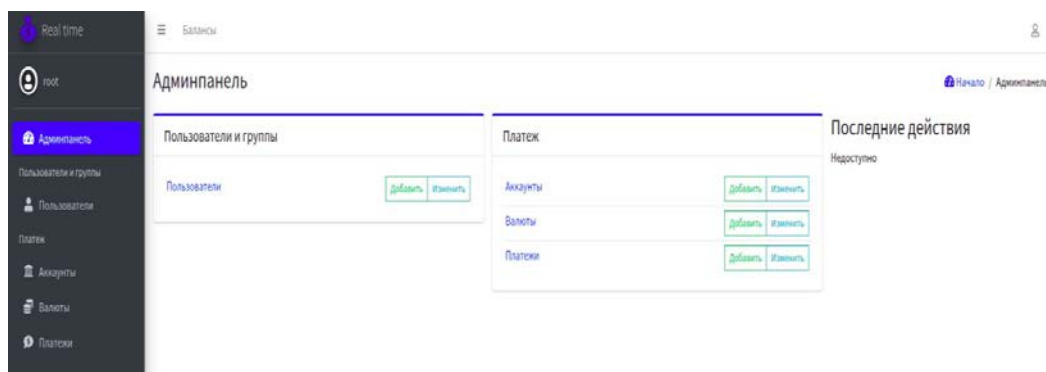


Рисунок 5 – главная страница

Данное приложение имеет большое количество направлений для расширения. Например, можно увеличить количество поддерживаемых типов сообщений. Также можно добавить дополнительный функционал оповещений, это поможет более быстро реагировать на движения средств, которые клиенты хотят отслеживать.

Список использованных источников:

1. Всяких, М.В. Современные платежные системы коммерческого банка /М.В. Всяких // Концепт. – 2015. – № 08 (август) [Электронный ресурс]. – Режим доступа: <http://e-koncept.ru/2015/15265.htm>. – Дата доступа: 02.04.2022
2. Буч, Г. Язык UML. Руководство пользователя. 2-е изд. / Г. Буч, Д. Рамбо, И. Якобсон. – М. : ДМК Пресс, 2006. – 496 с.