УДК 004.42:336.71

4. ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОРГАНИЗАЦИИ СИСТЕМЫ ОБСЛУЖИВАНИЯ КЛИЕНТОВ БАНКА В РЕЖИМЕ «ЕДИНОГО ОКНА»

Микшас П.В., студент гр. 072304

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

Сторожев Д.А. – ст. преподаватель каф. ЭИ

Аннотация. В данной статье рассматривается программное средство, разработанное для эффективной организации системы обслуживания клиентов банка в режиме «единого окна». Исследование предлагает комплексный подход к автоматизации банковских процессов, объединяя в себе инструменты оформления заявок, обработки заявок, проведения электронного платежа и аналитики данных. Разработанное программное средство обеспечивает оптимизацию времени обслуживания клиентов, повышение удовлетворенности клиентов и снижение нагрузки на персонал банка. Результаты исследования демонстрируют эффективность и перспективность применения предложенной системы в банковской сфере.

Ключевые слова. Информационная система, банковский бизнес, GUI-интерфейс, база данных, электронный банковский платеж, администратор, «Интернет-банкинг», клиент, разработка клиент-серверного приложения.

Информационно-банковские технологии – это та часть информационного пространства, которая используется для ведения банковской деятельности и предоставления банковских услуг. Традиционно общество относится к технологической новизне в консервативной банковской деятельности несколько предвзято, однако, внедрение информационно-банковских технологий в экономику – процесс положительный и полезный, стимулирующий развитие экономики и банковского дела по инновационному пути.

Обычно под "интернет-банкингом" понимают оказание услуг банками по дистанционному управлению счетом. Удаленное управление счетом через Интернет обычно подразумевает проверку состояния счета, оплату разнообразных счетов и перевод средств с одного счета на другой, а также предоставление клиенту информационной поддержки и многочисленных сопутствующих услуг.

По статистике более 80% всех банковских операций человек может делать, сидя за компьютером дома или в офисе. Выгода для банкиров и их клиентов очевидна: первые значительно сокращают издержки на содержание филиальной сети, а вторые получают дополнительные удобства [1].

«Интернет банкинг» прост в использовании. Нет необходимости обладать какими-либо особенными знаниями или навыками, чтобы управлять своими счетами через Интернет. Несмотря на явные преимущества, банки не спешат внедрять новые технологии. Главной причиной медленного распространения интернет-банкинга является относительная небезопасность расчетов и сохранности средств на счетах клиентов. Возможность несанкционированного доступа к чужой информации остается основной проблемой в Интернете.

Важным свойством безопасности «Интернет-банкинга» является подтверждение транзакций с помощью одноразовых паролей (чтобы перехват трафика не давал бы злоумышленнику возможности получить доступ к нашим финансам).

Таким образом, объект исследования – банковский бизнес и системы автоматизации оказания банковских услуг. Предмет исследования – программное обеспечение системы автоматизации оказания банковских услуг.

Для взаимодействия с программным средством существует 2 актера: администратор и пользователь. Как видно из диаграммы, набор доступных им возможностей отличается.

Так как у каждого пользователя «Интернет-банкинга» должен иметься свой счет и банковская карта, необходимо спроектировать базу данных, которая осуществляла бы их хранение в электронном виде. Это позволит клиенту облегчить проведение операций по платежам, оформлению новых банковских карт, открытию счетов, кредитов и вкладов. Для корректной работы необходимо предусмотреть возможность добавления, удаления и редактирования информации в базе данных. Для хранения информации будет использован MYSQL Server. Подключение к нему будет осуществляться при авторизации, и, в зависимости от полученных прав, пользователь, будет получать определённый набор возможностей для выполнения своей работы.

Также пользователь будет иметь возможность создавать заявки на получение кредита, создавать и управлять своими вкладами и счетами, просматривать журнал своих операций, просматривать курсы валют, а также видеть график изменения иностранных валют к белорусскому рублю. Поэтому все соответствующие таблицы будут добавлены в базу данных.

Диаграмма функциональных возможностей пользователя приведена на рисунке 1.

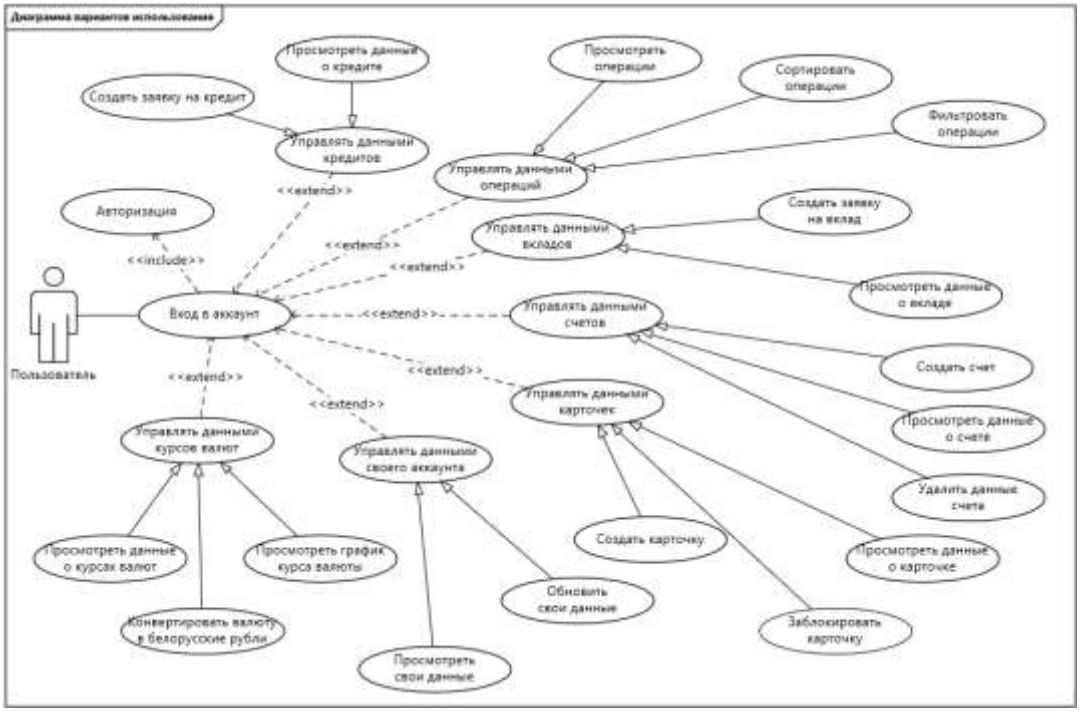

Рисунок 1 – Диаграмма вариантов использования пользователя

Функциональные возможности администратора более расширенные, чем у пользователя. Администратору не доступен журнал операций пользователя, так как это считается конфиденциальной информацией, которую может просматривать лишь пользователь, но администратор способен изменять статусы счетов, вкладов, кредитов, карточек. Также администратор не может создавать вклады, кредиты или заказывать банковские карты, так как он является лишь управляющим звеном. Администратор может регистрировать пользователей в системе и отправлять письмо с их логином и паролем им на почту. Также администратору доступна работа с заявками пользователей на открытие вкладов и кредитов. Администратор способен просматривать курсы валют и их графики, но он не может их изменять, так как данные берутся с официального сайта Национального банка Республики Беларусь. Диаграмма вариантов использования администратора представлена на рисунке 2.

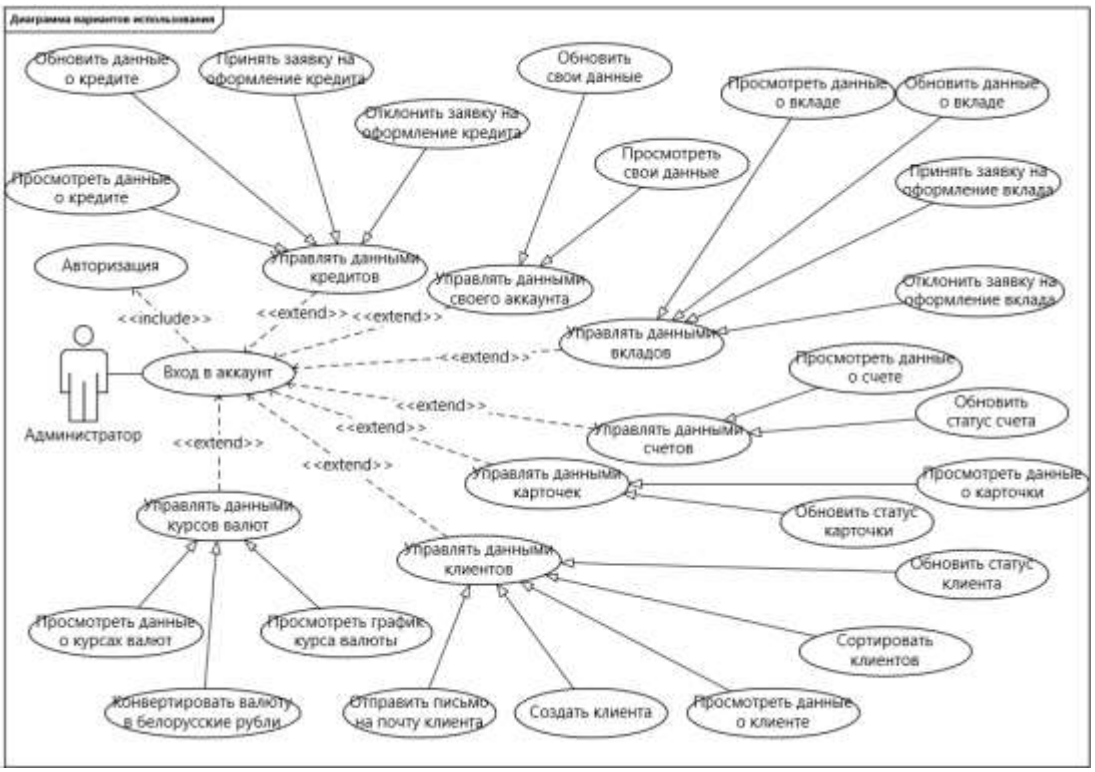

Рисунок 2 – Диаграмма вариантов использования администратора

Одним из требований к хорошему графическому интерфейсу программной системы является концепция «предсказуемости», чтобы система работала предсказуемо, чтобы пользователь заранее интуитивно понимал, какое действие выполнит программа после получения его команды.

Приложение имеет оконный интерфейс, разработанный с использованием библиотеки React языка JavaScript. Главная страница пользователя содержит вкладки, предоставляющие доступ к различным пунктам меню – история операций, вклады, кредиты, платежи и курсы валют. При входе в свой личный кабинет активна страница «Мой аккаунт». Все страницы спроектированы однотипно. В верхней части страницы располагается меню для быстрого перехода между страницами. Под меню находится основная контентная часть страницы, в которой располагаются соответствующие данные для каждой вкладки.

Основная часть страницы «Мой аккаунт» разделена на 2 части (рисунок 3). В левой части находятся основные данные пользователя такие, как его имя, фамилия, логин, дата рождения, почта и номер телефона. Также под данными располагается кнопка для редактирования данных. При ее нажатии откроется модальное окно с формой редактирования. В правой части отображаются счета пользователя. Под счетами находится кнопка для добавления нового счета, при нажатии на которую откроется модальное окно с формой для добавления нового счета. На счета также можно нажимать, тогда откроется новая страница с более подробными данными о счете, а также с картами, привязанными к выбранному счету. Кнопка «Выйти», расположенная в правом верхнем углу отвечает за выход из приложения.

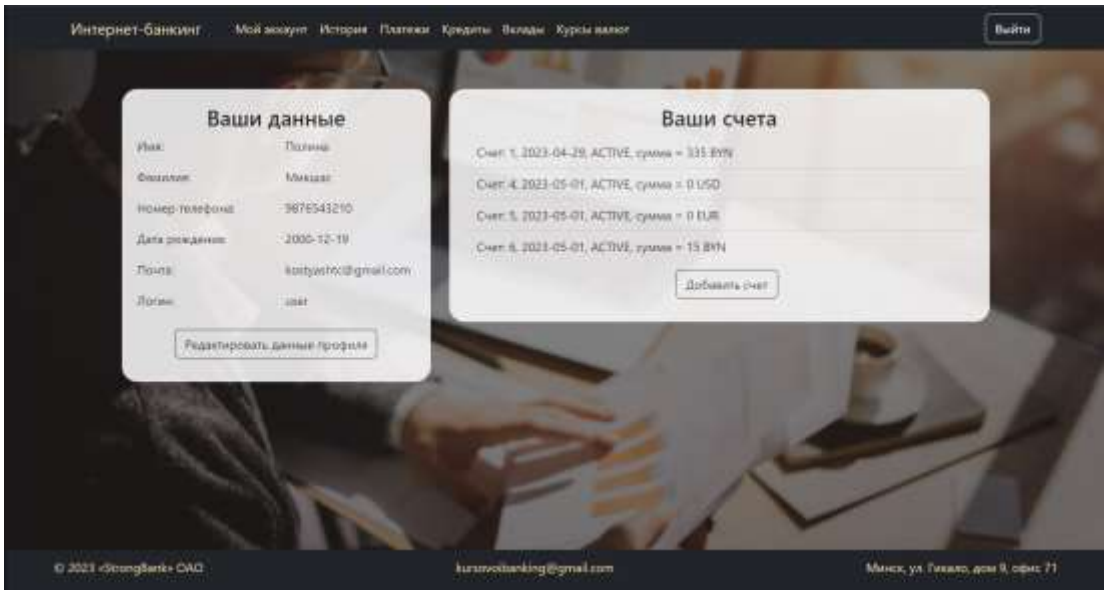

Рисунок 3 – Главное меню пользователя

При запуске программы появится страница аутентификации (проверка имени и пароля пользователя программы). Данная страницы служит для идентификации пользователя программы в целях предоставления ему определенных прав в работе с программой. После первой установки программы имеется только один пользователь – "Администратор". Он является полноправным владельцем всех данных, с которыми может взаимодействовать программа.

Диаграмма последовательности разрабатываемой системы представлена на рисунке 4. На представленной диаграмме можно увидеть действующих субъектов всей информационной системы (пользователь, клиент, сервер, база данных).

На диаграмме показан процесс оформления новой заявки (например, заявки для открытия нового вклада). Сперва пользователь заполняет необходимые поля в клиентской форме, после данные проверяются и отправляются на сервер в виде запроса на добавление новой заявки. После обработки запроса сервер создаёт SQL команду и посылает её в базу данных. В базе данных команда обрабатывается, и возвращается результат выполнения SQL запроса. Сервер обрабатывает ответ и посылает его обратно клиенту. Тот, в свою очередь, проверяет, успешно ли выполнен запрос, и выводит соответствующее сообщение на экран пользователю.

На диаграмме также можно увидеть «линию жизни». Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях.

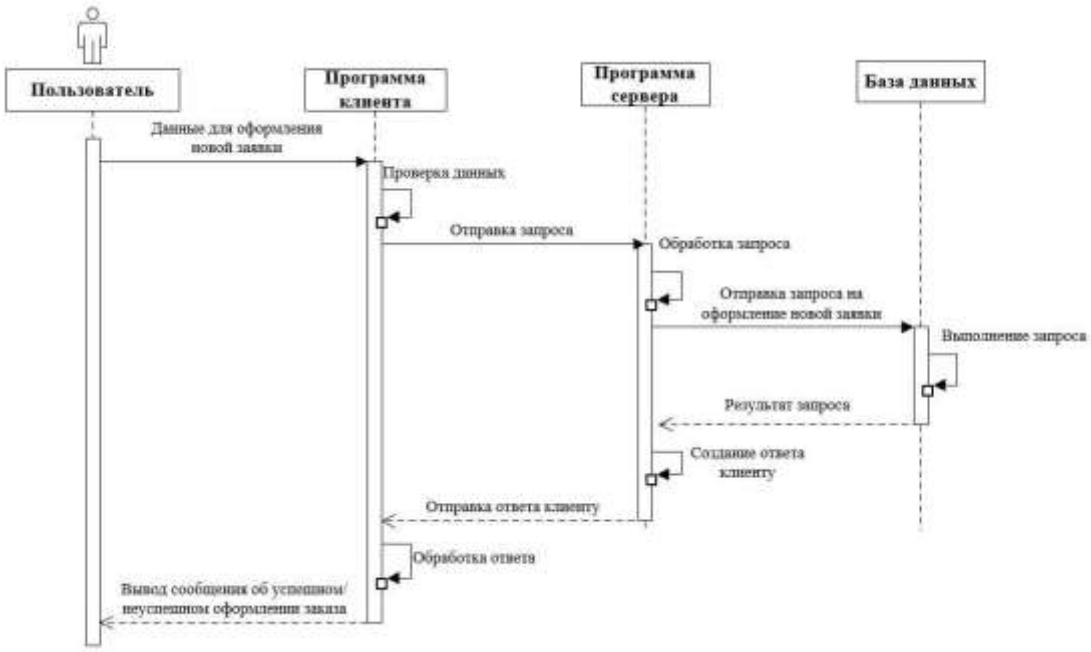

Рисунок 4 – Диаграмма последовательности оформления заявки

Для графического отображения структуры хранимых данных была использована методология IDEF1.X. IDEF1.X – язык моделирования данных для разработки семантики моделей данных. IDEF1.X используется для формирования графических представлений информационных моделей которые отражают структуру и семантику информации внутри среды или системы [2]. Для разрабатываемого программного средства понадобятся такие сущности как Клиент, Заявка на кредит, Кредит, Заявка на вклад, Вклад, Счет, Карточка, Операция, Роль. Схема структуры хранимых данных системы представлена на рисунке 5.

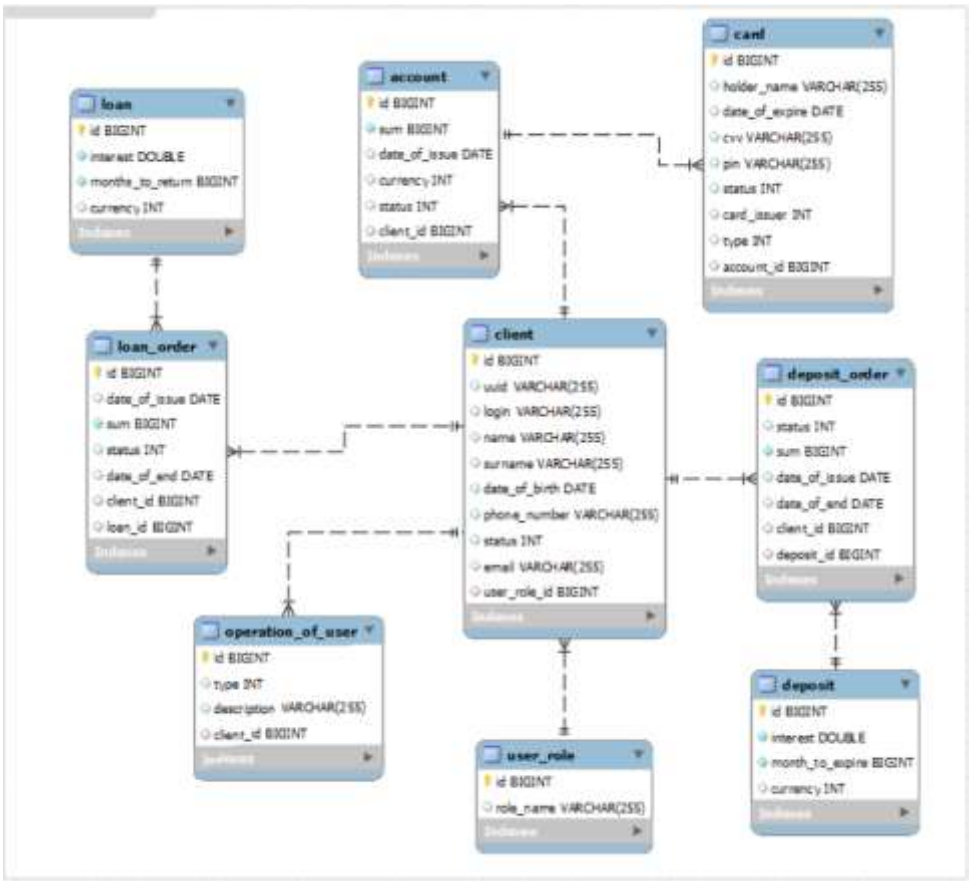

Рисунок 5 – Диаграмма IDEF1.X

В совсременном мире обеспечение безопасности данных играет важную роль в различных сферах деятельности. Существует множество угроз, связанных с хранением, обработкой и передачей данных, таких как хакерские атаки, вирусы, фишинг, мошенничество и многие другие. Для защиты данных и предотвращения таких угроз используются различные методы и средства безопасности, включая аутентификацию и авторизацию, шифрование данных, политику CORS и другие.

Политика CORS (Cross-Origin Resource Sharing) – это механизм безопасности, который используется в веб-браузерах для защиты пользователей от атак, связанных с запросами к веб-ресурсам из другого источника. Эта политика определяет, как браузеры должны ограничивать доступ к веб-ресурсам с разных источников [3].

Без политики CORS любой веб-сайт может отправлять запросы к любому другому веб-сайту. Это может привести к уязвимостям безопасности, так как это позволяет злоумышленникам получить доступ к конфиденциальным данным и взаимодействовать с другими веб-сайтами от имени пользователя.

Keycloak – это средство управления идентификацией и доступом (Identity and Access Management – IAM), которое обеспечивает аутентификацию и авторизацию пользователей в веб-приложениях. Keycloak предоставляет ряд функций, таких как управление пользователями, ролями, группами, настройку двухфакторной аутентификации, обработку забытых паролей, а также интеграцию с различными сторонними системами аутентификации, такими как LDAP, Active Directory и OAuth **Ошибка! Источник ссылки не найден.**.

Аутентификация – это процесс проверки подлинности пользователя, который запрашивает доступ к системе или приложению. Система аутентификации LDAP используется для проверки подлинности пользователей в веб-приложении. LDAP (Lightweight Directory Access Protocol) – это стандартный протокол для доступа к директориям, в которых хранится информация об учетных записях пользователей, группах и других объектах. При использовании системы LDAP, пользователи должны вводить свой логин и пароль для получения доступа к приложению. LDAP также используется для хранения и управления учетными записями пользователей и группами.

Авторизация – это процесс определения прав доступа пользователя после успешной аутентификации. Протокол OAuth2.0 используется для авторизации пользователей в веб-приложении. OAuth2.0 – это протокол авторизации, который позволяет пользователям давать доступ к своим ресурсам, например, календарю или контактам, другим приложениям без необходимости раскрытия своих учетных данных. При использовании OAuth2.0, приложения запрашивают доступ к ресурсам пользователя через сервер авторизации, который затем возвращает токен авторизации, который приложение может использовать для доступа к ресурсам пользователя.

В разрабатываемом программном средстве используется аутентификация LDAP для проверки подлинности пользователей и авторизация OAuth2.0 для определения и управления их правами доступа. Эти системы помогают обеспечить безопасность доступа к данным и функциональности, предотвращая несанкционированный доступ к приложению и обеспечивая защиту данных пользователей.

JWT (JSON Web Token) – это стандартный формат токена, используемый для передачи данных между сторонами в форме JSON объекта. Токен состоит из трех частей: заголовка, полезной нагрузки (payload) и подписи [4].

В данной статье была рассмотрена информационная система, способная облегчить работу пользователя и увеличить её эффективность за счёт экономии времени и усилий. На клиентской части реализован графический интерфейс, значительно упрощающий использование приложения рядовому сотруднику. Интерфейс выполнен в максимально минималистичном стиле, что способствует облегчению процесса обучения персонала для использования данной системы.

Взаимодействие приложения с базой данных обеспечивает надёжность и структурированность хранения всей информации. Клиент-серверная архитектура позволяет контролировать целостность данных и снижать нагрузку на сеть путём разделения общего процесса на задачи. Источником таких запросов является клиент, а сервер осуществляет непосредственное взаимодействие с базой данных с целью получения запрашиваемой информации. Важной особенностью программы является лёгкая модифицируемость, в дальнейшем это поможет вносить как незначительные, так и большие изменения модификации.

Список использованных источников:

1. Лаврушин, О. И. Мировые тенденции развития банковской деятельности и банковских технологий / О.И. Лаврушин.– М.: Финансы и статистик, – 2023.

2. Методология IDEF1.X [Электронный ресурс]. – Режим доступа: https://studme.org/ekonomika/metodologiya_idef1x.– Дата доступа: 27.03.2024.

3. CORS [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Web/HTTP/CORS. – Дата доступа: 27.03.2024.

4. JSON Web Tokens (JWT) [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/340146/. – Дата доступа: 27.03.2024.