РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет путей сообщения» (СГУПС)
кафедра «Информационные технологии транспорта»

Научно-исследовательская работа на тему «Разработка мобильного приложения для определения мошеннических транзакций в банках»
Вид практики: преддипломная

Проверили:
ст. преподаватель

_____ А. А. Уланов
(подпись)

_____
(дата проверки)

Выполнил:
студент гр. БПИ-411

_____К. В. Рязанов
(подпись)

_____
(дата сдачи на проверку)

Краткая рецензия:

_____

_____

_____

_____

_____

_____
(запись о допуске к защите)

_____ _____
(оценка по результатам защиты) (подписи преподавателей)

_____
(дата защиты)

Новосибирск
2025

УДК 004.41

АННОТАЦИЯ

В отчете о преддипломной практике 42 страница, 12 рисунков, 17 источников.

Ключевые слова: *мобильный банкинг, мошеннические транзакции, антифрод-система, машинное обучение, мобильное приложение, Android, Kotlin, ASP.NET Core, уведомление пользователя, пользовательский опыт, прототип, преддипломная практика.*

Предметная область – проектирование и разработка прототипа мобильного приложения для демонстрации улучшенных механизмов информирования пользователей о рисках мошеннических транзакций в мобильном банкинге. Основная функция разрабатываемого прототипа заключается в проактивном уведомлении пользователя о потенциальной угрозе до завершения операции, наглядной визуализации уровня риска с использованием результатов анализа (включая методы машинного обучения) и предоставлении интуитивно понятных средств для подтверждения или отклонения подозрительной транзакции. Прототип предназначен для демонстрации концепции и потенциального улучшения существующих мобильных банковских систем. Разработка данного прототипа в рамках преддипломной практики направлена на повышение безопасности пользователей мобильного банкинга и улучшение их опыта взаимодействия с антифрод-системами.

ABSTRACT

The work contains 42 pages, 12 figures, 17 sources.

Keywords: mobile banking, fraudulent transactions, anti-fraud system, machine learning, mobile application, Android, Kotlin, ASP.NET Core, User notification, user experience, prototype, pre-graduate practice.

The subject area is the design and development of a prototype mobile application to demonstrate improved mechanisms for informing users about the risks of fraudulent transactions in mobile banking. The main function of the prototype being developed is to proactively notify the user of a potential threat before the operation is completed, visually visualize the risk level using analysis results (including machine learning methods), and provide intuitive tools to confirm or reject a suspicious transaction. The prototype is intended to demonstrate the concept and potential improvement of existing mobile banking systems. The development of this prototype as part of a pre-graduate internship is aimed at improving the security of mobile banking users and improving their experience of interacting with anti-fraud systems.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Антифрод-система (Anti-Fraud Engine) – Система, предназначенная для выявления и предотвращения мошеннических операций в режиме реального времени или постфактум, часто с использованием анализа данных, правил и алгоритмов машинного обучения.

Машинное обучение (ML) – Класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

Мобильное приложение – Программное обеспечение, разработанное для работы на мобильных устройствах, таких как смартфоны и планшеты.

Мошенническая транзакция – Финансовая операция (перевод, платеж), совершенная злоумышленником без ведома или согласия легитимного владельца счета с целью хищения денежных средств или получения несанкционированного доступа к информации.

Пользовательский интерфейс (UI / GUI) – Совокупность средств (экраны, кнопки, меню), при помощи которых пользователь взаимодействует с программной системой.

Прототип – Ранняя, часто упрощенная, рабочая версия программного продукта, созданная для демонстрации основной концепции, проверки гипотез, сбора обратной связи или тестирования ключевого функционала.

Сервис-ориентированный подход (SOA) – Архитектурный стиль построения распределенных систем, при котором функциональность представлена в виде независимых, слабосвязанных сервисов, взаимодействующих друг с другом по сети.

Социальная инженерия – Метод получения несанкционированного доступа к информации или системам, основанный на использовании психологических манипуляций людьми, а не на технических взломах.

Транзакция – Любая операция, связанная с движением денежных средств по счету.

Фишинг – Вид интернет-мошенничества, заключающийся в создании поддельных веб-сайтов, рассылке электронных писем или сообщений от имени банков или других организаций с целью выманивания у пользователей конфиденциальной информации.

API (Application Programming Interface) – Программный интерфейс приложения; набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах.

ASP.NET Core – Кроссплатформенный фреймворк с открытым исходным кодом для создания современных веб-приложений и API, разработанный Microsoft.

Backend (Серверный модуль) – Программно-аппаратная часть сервиса, отвечающая за его внутреннюю логику, обработку данных, взаимодействие с базой данных и другими системами.

Frontend (Клиентский модуль) – Пользовательский интерфейс; часть сервиса, с которой непосредственно взаимодействует пользователь.

Kotlin – Статически типизированный язык программирования, работающий поверх виртуальной машины Java (JVM) и официально поддерживаемый Google для разработки приложений под ОС Android.

PostgreSQL – Свободная объектно-реляционная система управления базами данных (СУБД).

Visual Studio – Интегрированная среда разработки (IDE), разрабатываемая корпорацией Microsoft.

Android Studio – Официальная интегрированная среда разработки (IDE) для платформы Android.

UML (Unified Modeling Language) – Унифицированный язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

ПЕРЕЧЕНЬ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

ИС – информационная система.

.NET – программная платформа компании Microsoft.

MS – компания Microsoft.

IDE – интегрированная среда разработки.

ACID – Atomicity, Consistency, Isolation, Durability (Атомарность, Согласованность, Изолированность, Долговечность)

API – Application Programming Interface (Программный интерфейс приложения)

CPU – Central Processing Unit (Центральный процессор)

GUI – Graphical User Interface (Графический интерфейс пользователя)

ML – Machine Learning (Машинное обучение)

ОС – Операционная система

ПО – Программное обеспечение

SDK – Software Development Kit (Комплект для разработки программного обеспечения)

SQL – Structured Query Language (Язык структурированных запросов)

UI – User Interface (Пользовательский интерфейс)

UML – Unified Modeling Language (Унифицированный язык моделирования)

ВТБ – Банк ВТБ (используется как пример)

ЕСИА – Единая система идентификации и аутентификации

ЖКХ – Жилищно-коммунальное хозяйство

ИНН – Идентификационный номер налогоплательщика

НИУ ВШЭ – Национальный исследовательский университет «Высшая школа экономики»

РФ – Российская Федерация

СУБД – Система Управления Базами Данных

ФНС – Федеральная налоговая служба

ЦБ РФ – Центральный банк Российской Федерации

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современную эпоху цифровой трансформации мобильные банковские приложения стали неотъемлемым инструментом для управления финансами, обеспечивая пользователям удобный и быстрый доступ к широкому спектру банковских услуг. Стремительное увеличение объема мобильных операций и повсеместное проникновение смартфонов в повседневную жизнь сделали их ключевым каналом взаимодействия между банками и клиентами. Однако этот прогресс сопровождается беспрецедентным ростом и усложнением мошеннических действий в данной сфере. Злоумышленники активно используют все более изощренные методы, такие как фишинг, социальная инженерия, распространение вредоносных приложений и даже применение DeepFake-технологий, превращая мобильные платформы в основную цель для атак [1-3].

Финансовые институты и регуляторные органы предпринимают значительные усилия для противодействия этим угрозам. Современные российские банки, такие как СберБанк, ВТБ и Т-Банк, активно внедряют комплексные антифрод-системы, сочетающие алгоритмы машинного обучения, поведенческий анализ и многофакторную аутентификацию [7-10]. Центральный банк РФ также усиливает контроль посредством таких инициатив, как платформа «Цифровой след» [7]. Тем не менее, несмотря на демонстрируемую эффективность существующих систем, сохраняется ряд ограничений: их работа зачастую носит реактивный характер (реагирование постфактум), недостаточна персонализация проверок, а механизмы уведомления и взаимодействия с пользователем не всегда оптимальны, порой вызывая раздражение из-за частых верификаций [13]. Это подчеркивает актуальность поиска новых подходов к защите клиентов.

Целью настоящей выпускной квалификационной работы является разработка прототипа мобильного приложения, демонстрирующего улучшенные механизмы взаимодействия пользователя с рисками

мошеннических транзакций. Основная идея заключается в создании решения, которое не просто блокирует подозрительные операции, а проактивно и наглядно информирует пользователя о потенциальных угрозах до момента завершения транзакции, предоставляя ему возможность осознанного принятия решения и повышая его осведомленность. Приложение будет использовать симуляцию работы антифрод-системы на основе анализа транзакций (с потенциальным использованием ML-моделей) и фокусироваться на интуитивно понятном интерфейсе, мгновенных, но не навязчивых уведомлениях и элементах интерактивного обучения.

Для достижения поставленной цели в работе решаются следующие задачи:

- Провести анализ предметной области – выявления и предотвращения мошеннических транзакций в мобильном банкинге.
- Исследовать существующие аналоги антифрод-систем и подходов к уведомлению пользователей в ведущих российских банках.
- Обосновать актуальность разработки прототипа с фокусом на улучшение пользовательского опыта и проактивное информирование.
- Спроектировать архитектуру информационной системы, включая моделирование бизнес-процессов взаимодействия пользователя с системой.
- Разработать компоненты прототипа: серверную часть (Backend) на ASP.NET Core 8, клиентскую часть (мобильное приложение) для Android на Kotlin, структуру базы данных на PostgreSQL 16.
- Продемонстрировать работу прототипа по симуляции анализа транзакций и информированию пользователя о рисках.

Объектом исследования является процесс взаимодействия пользователя с мобильным банковским приложением в контексте выявления и предотвращения мошеннических транзакций.

Предметом исследования выступают методы и средства проектирования и разработки информационной системы (прототипа мобильного приложения), направленной на улучшение информированности пользователя о мошеннических рисках.

Работа состоит из введения, трех глав, заключения и списка использованных источников. В первой главе проводится аналитическое исследование предметной области, аналогов и обосновывается актуальность разработки. Во второй главе описывается процесс проектирования информационной системы: моделирование бизнес-процессов, определение структуры ИС и выбор средств разработки, а также формулируются требования к системе. Третья глава посвящена описанию этапов разработки прототипа: созданию базы данных, серверной и клиентской частей приложения, а также их интеграции. В заключении подводятся итоги проделанной работы.

1 Аналитическое исследование

1.1 Описание предметной области

Предметная область исследования посвящена процессу выявления и предотвращения мошеннических транзакций в рамках мобильных банковских приложений. Данная тематика приобретает особую значимость ввиду широкого распространения цифровых технологий и стремительного увеличения объема мобильных банковских операций, что сопровождается существенным ростом числа мошеннических действий в этой сфере.

Современная ситуация демонстрирует значительные изменения в характере и масштабе мошенничества. Динамичное развитие цифровых платформ и активное проникновение мобильных технологий делают смартфоны ключевым каналом оказания банковских услуг, одновременно превращая их в основную цель для злоумышленников. Рассмотрим подробно характерные черты современных угроз и способы их преодоления.

Одной из главных характеристик текущего периода является активизация кибермошенников, использующих разнообразные приемы для завладения средствами клиентов банков. Наиболее распространенные виды мошенничества включают фишинг и социальную инженерию. Практика показывает, что около 45% мошеннических атак реализуются именно такими способами [1].

Суть фишинга заключается в создании поддельных сайтов и рассылке писем или SMS-сообщений, имитирующих официальное обращение банка. Пользователи вводят персональные данные на таких страницах, полагая, что взаимодействуют непосредственно с банком, однако впоследствии их данные попадают в руки злоумышленников.

Дополнительную опасность представляют вредоносные приложения, распространяемые через сторонние магазины или рассылки спама. Они маскируются под оригинальные банковские продукты, предлагая установить программу, похожую на оригинальное приложение банка. Однако после

установки такие приложения собирают всю необходимую информацию, такую как пароли, номера карт и коды подтверждения, позволяя преступникам проводить несанкционированные платежи [2].

В последнее время значительное распространение получили техники, основанные на глубоких нейронных сетях (так называемые DeepFake-технологии). Эти методы позволяют злоумышленникам реалистично воспроизводить внешность и речь реальных людей, включая сотрудников банка. Подделанные звонки и сообщения создают иллюзию общения с представителем кредитной организации, вводя жертву в заблуждение относительно правомерности запрашиваемых действий [3].

Противодействие мошенническим действиям включает комплекс мероприятий организационного и технологического характера. Организационные меры предполагают повышение уровня информированности клиентов о существующих видах мошенничества и формирование привычек осторожного пользования мобильными устройствами. Многие крупные российские банки проводят регулярные информационные кампании, направленные на обучение клиентов правилам безопасной работы с мобильными приложениями и предупреждения о возможностях мошенничества.

Важнейшую роль играют специализированные антивирусные программы и системы поведенческого анализа. Последние работают на основе алгоритмов машинного обучения, способных выявить нестандартные паттерны поведения пользователя и предупредить о потенциальной опасности транзакции. Особенно эффективны такие системы в случаях массовой рассылки сообщений с предложением перевести средства на подставные счета или предоставить секретные данные [4].

Регуляторные органы предпринимают шаги по ужесточению правил ведения банковской деятельности в отношении удаленных каналов обслуживания. Центробанк России выпустил серию рекомендаций, предусматривающих обязательную интеграцию с биометрическими

системами для высоких уровней риска, периодический аудит используемых алгоритмов и реализацию стандартов защиты на основании федерального закона №115-ФЗ [5, 6].

Несмотря на усилия правоохранительных органов и самого банковского сообщества, угроза мошенничества продолжает оставаться серьезной проблемой. Высокий уровень технической оснащенности и развитость инфраструктуры интернета способствуют росту масштабов таких правонарушений. Важно отметить, что своевременное реагирование и принятие превентивных мер помогают предотвратить значительную долю негативных последствий.

Таким образом, понимание природы и динамики развития мошенничества в финансовой сфере имеет решающее значение для разработки эффективного инструментария защиты. Дальнейшее изучение этой темы позволит предложить практические рекомендации по совершенствованию механизмов защиты клиентов банков и повышению эффективности контрольных функций в сфере электронной коммерции.

1.2 Анализ аналогов

Современные российские банки активно внедряют системы обнаружения мошеннических транзакций, которые сочетают алгоритмы машинного обучения, поведенческий анализ и многофакторную аутентификацию. В рамках исследования был проведен анализ решений, реализованных в мобильных приложениях ведущих банков России, таких как СберБанк, ВТБ и Т-Банк, а также рассмотрены регуляторные инициативы.

В 2024 году Центральный банк Российской Федерации запустил платформу мониторинга транзакций «Цифровой след», которая интегрирует данные из 320 кредитных организаций. Эта система выявляет аномалии, такие как массовая регистрация карт на фиктивные лица, что уже привело к обнаружению 8,4 тыс. таких случаев с общим оборотом 3,2 млрд рублей [7]. Для анализа связей между счетами используются графовые алгоритмы, а

результаты обработки передаются в банки для блокировки подозрительных операций.

В мобильном приложении СберБанка реализован модуль AI Fraud Detection, который анализирует более 150 параметров транзакции, включая геолокацию, скорость набора кода и историю операций. При обнаружении аномалий, например, при переводе в новый регион, система приостанавливает операцию и отправляет push-уведомление с запросом подтверждения через голосовой биометрический шаблон [8]. Это позволяет оперативно реагировать на подозрительные действия и защищать клиентов.

ВТБ использует систему реального времени VTB Anti-Fraud, которая сопоставляет транзакции с паттернами мошенничества из базы FICO Falcon. Для операций с высоким риском, таких как смена реквизитов получателя, активируется автоматический звонок клиенту от робота с синтезированным голосом, имитирующим сотрудника службы безопасности [9]. Это решение позволяет быстро информировать клиентов о потенциальных угрозах и предотвращать мошенничество.

Т-Банк внедрил адаптивную аутентификацию на основе оценки риска транзакции. Для переводов до 10 тыс. рублей достаточно SMS-кода, в то время как для более крупных сумм требуется подтверждение через Face ID. В 2023 году это нововведение снизило число ложных блокировок на 27% при сохранении уровня безопасности [10], что свидетельствует о высоком уровне эффективности системы.

С учетом новых регуляторных требований, таких как директива PSD3, российские банки адаптируют свои системы под эти нормы, внедряя автоматический запрос ИНН контрагента через API ФНС [11]. Это позволяет улучшить контроль за транзакциями и повысить уровень безопасности.

Анализ показал, что существует тенденция к балансу между безопасностью и удобством для пользователей. Например, Тинькофф Банк использует «динамическую биометрию», запрашивая отпечаток пальца только при отклонении транзакции от типового поведения клиента. Это

значительно сокращает среднее время подтверждения до 4,7 секунды [12]. Однако, согласно опросу НИУ ВШЭ, 68% пользователей отмечают, что частые запросы на верификацию вызывают раздражение, что требует тонкой настройки порогов срабатывания [13].

1.3 Актуальность разработки

Существующие системы демонстрируют свою эффективность, снижая уровень мошенничества на 18–34% по данным Центрального банка [7]. Однако они имеют определенные ограничения, такие как зависимость от реактивных методов, недостаточная персонализация проверок и слабая интеграция с внешними источниками данных, например, с биометрией ЕСИА. Это подчеркивает необходимость разработки предиктивного решения, которое будет проактивно уведомлять клиентов о подозрительных транзакциях.

Целью данной работы балы выбрана разработка мобильного приложения которое бы демонстрировало как текущие банковские приложения могли бы улучшить работу пользователей с рисками мошеннических транзакций, которое будет использовать алгоритмы машинного обучения для анализа транзакций. Это решение направлено на устранение недостатков существующих систем, обеспечивая более доступный и удобный способ для пользователей и банков. Основной задачей является создание интуитивно понятного интерфейса, который позволит пользователям легко взаимодействовать с приложением и получать уведомления о подозрительных транзакциях.

Важность разработки мобильного приложения для определения мошеннических транзакций заключается не только в повышении уровня безопасности, но и в улучшении пользовательского опыта. Современные клиенты ожидают от банков не только надежности, но и удобства в использовании услуг. Мобильное приложение, которое предоставляет пользователям возможность самостоятельно контролировать свои

транзакции и получать мгновенные и не навязчивые уведомления о подозрительных действиях, а так же интерактивное обучение пользователей по работе с новым функционалом, может значительно повысить доверие к финансовым учреждениям. Кроме того, успешная реализация данного проекта может стать основой для дальнейших исследований в области финансовых технологий.

Таким образом, разработка мобильного приложения для определения мошеннических транзакций представляет собой важный шаг в направлении повышения безопасности и удобства банковских услуг. Это решение не только отвечает на актуальные вызовы, стоящие перед финансовыми учреждениями, но и открывает новые горизонты для дальнейших исследований и разработок в области защиты от мошенничества.

2 Проектирование информационной системы

2.1 Моделирование бизнес-процессов ИС

Для построения диаграммы вариантов использования, описывающей процесс взаимодействия пользователя с мобильным приложением, на диаграмму был помещен актор — пользователь [14]. В результате, на диаграмме были добавлены варианты использования, такие как «Совершение транзакции», «Создание карты» и «Блокировка карты». Эти варианты использования были соединены с актором связью «Association», что позволяет отразить взаимодействие пользователя с системой, рисунок 2.1.

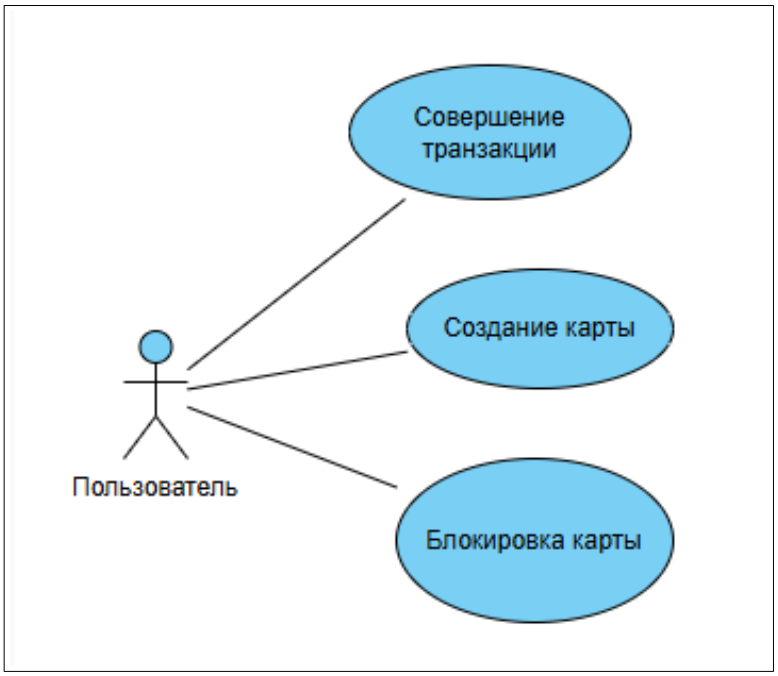

Рисунок 2.1 — Диаграмма вариантов использования

Далее были добавлены сценарии, относящиеся к мобильному приложению, включая «Отправить данные о транзакции», «Получить данные о проверке транзакции», «Отобразить пользователю результаты проверки транзакции», «Отобразить пользователю push-уведомление» и «Послать запрос на блокировку карты». Эти сценарии также были соединены с

мобильным приложением связью «Association», что демонстрирует их функциональную связь, рисунок 2.2 [15].

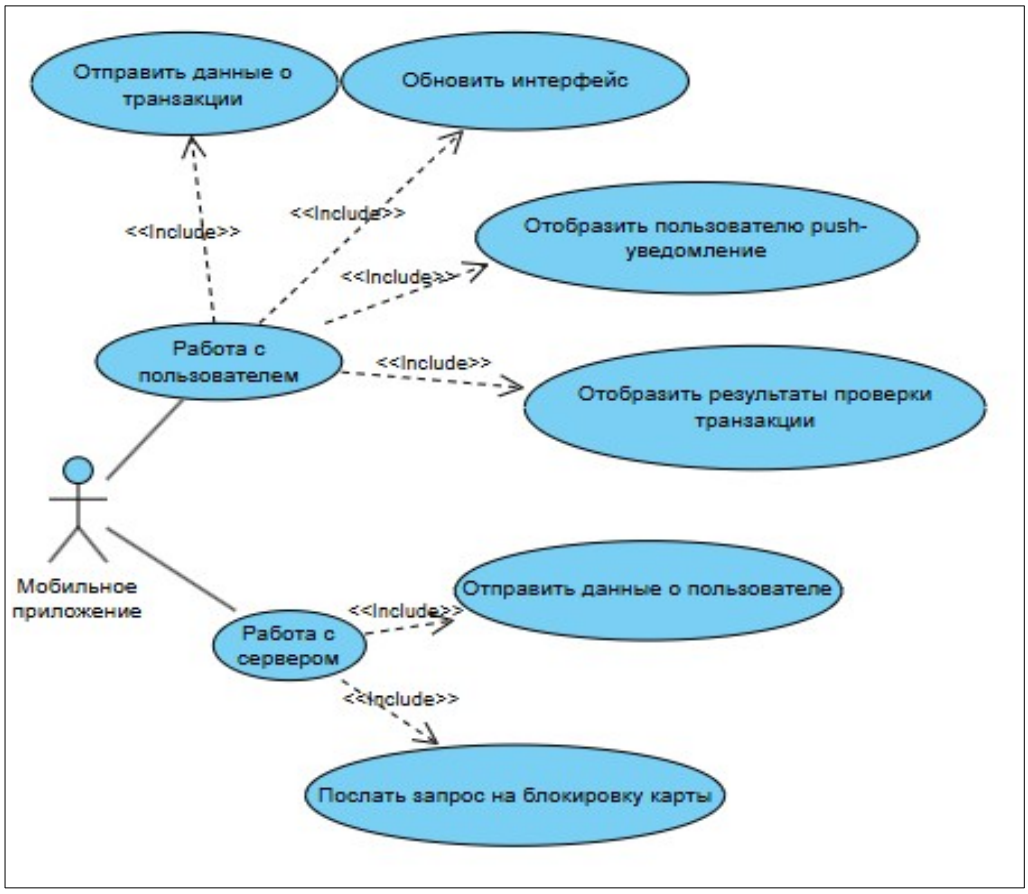

Рисунок 2.2 — Добавление сценариев для мобильного приложения

Также были добавлены сценарии, относящиеся к Антифрод-системе, включая «Получить данные транзакции», «Проверить транзакцию», «Отобразить пользователю результаты проверки транзакции» и «Отослать данные о проверки». Эти сценарии также были соединены с актором связью, рисунок 2.3.

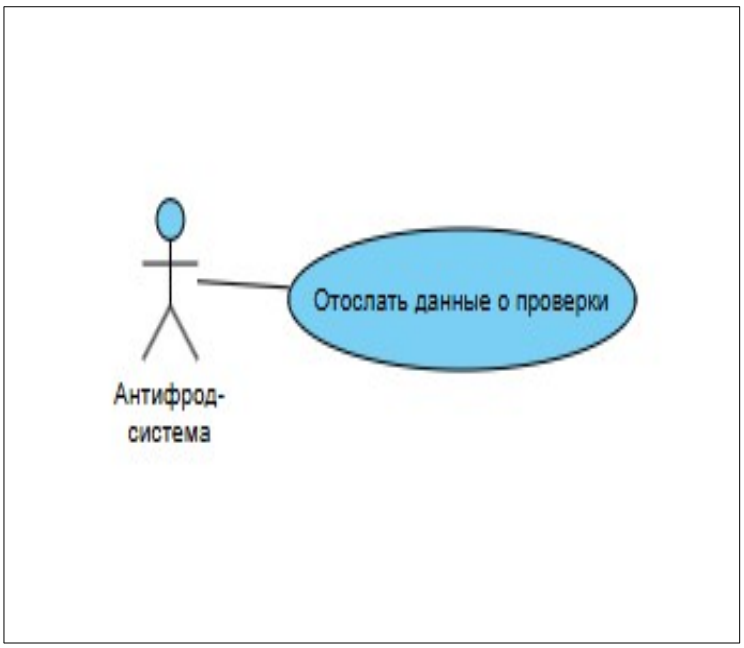

Рисунок 2.3 — Добавление сценариев для Антифрод-системы

Так же была создана диаграмма вариантов использования для серверного приложения с такими сценариями как, «Отправить данные пользователя», «Получить данные пользователя» и «Заблокировать карту», рисунок 2.4

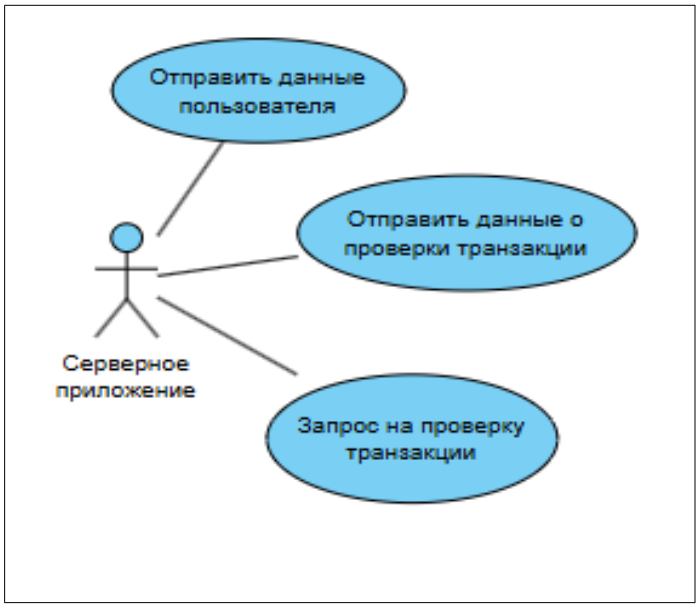

Рисунок 2.4 — Добавление сценариев для серверного приложения

Для построения диаграммы последовательности, описывающей процессы взаимодействия пользователя с системой при обработке транзакций и других операциях, на диаграмму был помещен актор — «Пользователь». Также были добавлены линии жизни объектов, представляющих ключевые компоненты системы: «Мобильное приложение», «Серверное приложение» и «Антифрод-система», рисунок 2.5.

Далее были добавлены последовательные сообщения между актором и объектами для иллюстрации основных сценариев взаимодействия, представленных на диаграмме:

- Сценарий создания карты: Пользователь инициирует создание карты (сообщение 1), мобильное приложение отправляет данные на сервер (1.1), сервер подтверждает получение или возвращает данные (1.2), интерфейс пользователя обновляется (1.3).
- Сценарий совершения транзакции и ее проверки: Пользователь инициирует транзакцию (2), мобильное приложение передает данные на сервер (2.1), серверное приложение запрашивает проверку у антифрод-системы (2.2), антифрод-система возвращает результат проверки (2.3), серверное приложение передает результат в мобильное приложение (2.4), которое отображает его пользователю (2.5).
- Сценарий блокировки карты: Пользователь инициирует блокировку (3.1), мобильное приложение посылает запрос на сервер (3.2), сервер обрабатывает запрос и может вернуть данные (3.3), интерфейс пользователя обновляется (3.4).

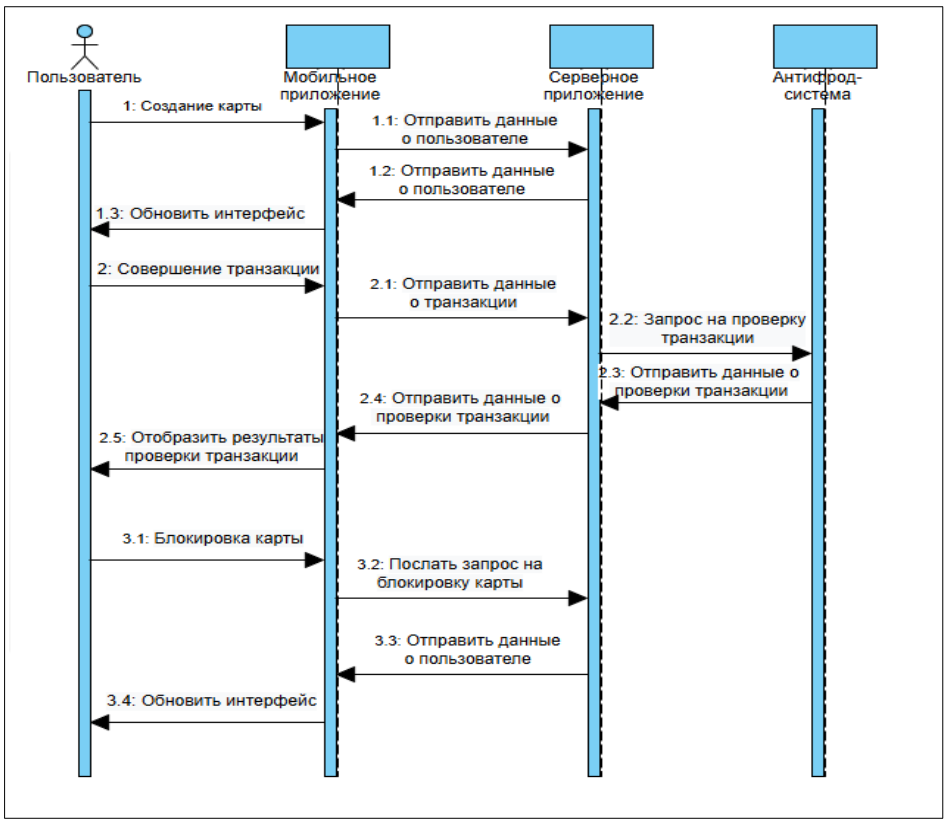

Рисунок 2.5 — Добавление последовательных сообщений

2.2 Структура ИС и ее средства разработки

Для достижения поставленных целей система спроектирована с использованием сервис-ориентированного подхода и включает следующие ключевые компоненты, взаимодействующие между собой для обеспечения полного цикла обработки транзакции и оценки ее риска:

Серверный модуль (Backend), является ядром системы, ответственным за прием и обработку запросов от клиентского мобильного приложения. Его функции включают:

- Аутентификацию и авторизацию пользователей.
- Валидацию входящих данных о транзакциях.
- Оркестрацию взаимодействия с другими компонентами системы (базой данных, сервисом антифрода).
- Формирование и отправку команд на отображение уведомлений в клиентское приложение.

- Логирование операций для последующего анализа и аудита.
- Предоставление API (Application Programming Interface) для клиентского приложения.

Клиентский модуль (Frontend / Мобильное приложение): Разрабатываемое нативное мобильное приложение для платформы Android. Предоставляет пользователю графический интерфейс для:

- Инициирования банковских транзакций (в рамках прототипа – симуляция).
- Просмотра истории операций.
- Получения и отображения push-уведомлений о результатах проверки транзакций, с четкой визуализацией уровня риска (например, цветовая индикация, понятные текстовые сообщения).
- Интерактивного взаимодействия с системой в случае обнаружения высокого риска.
- Отображения обучающих материалов или подсказок по безопасному поведению.

Модуль анализа и определения мошенничества (Anti-Fraud Engine): Специализированный сервис, реализующий логику выявления мошеннических транзакций.

Модуль базы данных (Database), представляет систему управления базами данных (СУБД), предназначенная для надежного и персистентного хранения всей необходимой информации:

- Профили пользователей (обезличенные данные для прототипа).
- История транзакций со всеми релевантными атрибутами (ананимизированные в прототипе)

Для эффективной реализации описанных компонентов информационной системы был тщательно подобран технологический стек,

призванный обеспечить необходимый уровень производительности, возможности масштабирования, безопасность и удобство разработки в рамках поставленной задачи по созданию прототипа для демонстрации улучшенного информирования пользователей о мошеннических транзакциях. В качестве основы для серверного компонента (Backend) был выбран ASP.NET Core версии 8.0.

Это современный, кроссплатформенный фреймворк от Microsoft, предназначенный для создания веб-приложений и API. Выбор пал именно на .NET 8 как на последнюю версию с долгосрочной поддержкой (LTS) на момент проектирования, что гарантирует не только высокую производительность и актуальные улучшения в области безопасности, но и длительный период поддержки со стороны разработчика. Технически, ASP.NET Core 8 работает на среде выполнения .NET Runtime, поддерживает языки программирования C# и F#, а в качестве встроенного веб-сервера использует Kestrel. Функциональные возможности этого фреймворка идеально подходят для нашего проекта: он позволяет легко создавать RESTful API с помощью ASP.NET Core Web API, что является стандартным подходом для взаимодействия с мобильным приложением в современных распределенных системах.

Критически важной является встроенная поддержка асинхронных операций (async/await), которая позволяет эффективно обрабатывать множество одновременных запросов от пользователей и взаимодействовать с базой данных или сервисом машинного обучения без блокировки потоков, обеспечивая тем самым высокую отзывчивость всей системы. Управление компонентами системы, такими как подключение к базе данных или вызов антифрод-модуля, а также их тестирование упрощается благодаря встроенному механизму внедрения зависимостей (Dependency Injection). Гибкость настройки конвейера обработки запросов обеспечивается через Middleware Pipeline, позволяя добавлять модули для аутентификации, авторизации, логирования и обработки ошибок. Кроме того, фреймворк

предоставляет встроенные механизмы для обеспечения безопасности API, включая поддержку JWT-токенов и настройку CORS-политик.

Обоснованием выбора ASP.NET Core служат его высокая производительность, развитая экосистема, превосходная интеграция со средой разработки Visual Studio, кроссплатформенность, дающая возможность развертывания на Windows, Linux или macOS, а также активное сообщество и надежная поддержка со стороны Microsoft.

Переходя к клиентской части системы (Frontend), для разработки мобильного приложения под платформу Android был выбран Kotlin SDK. Kotlin представляет собой современный, статически типизированный язык программирования, который официально поддерживается Google для Android-разработки. Он используется в связке с Android SDK (Software Development Kit), предоставляющим все необходимые библиотеки и инструменты. При разработке мы ориентируемся на Android API Level 34 (Android 14), обеспечивая при этом минимальную поддержку API Level 24 (Android 7.0) для охвата большинства активных устройств. Технически, Kotlin компилируется в байт-код JVM, что обеспечивает совместимость с Java, или может компилироваться в нативный код при использовании Kotlin Multiplatform, а для сборки проекта применяется система Gradle.

Выбор Kotlin обоснован тем, что это официальный язык для разработки под Android, он значительно повышает продуктивность разработчика, обеспечивает большую безопасность кода по сравнению с Java благодаря таким возможностям, как null safety, и активно развивается, предлагая современные подходы к разработке, например, корутины для асинхронных операций.

Для хранения данных, включая информацию о транзакциях, пользователях и результатах анализа рисков, была выбрана система управления базами данных PostgreSQL версии 16. Это мощная, объектно-реляционная СУБД с открытым исходным кодом. Мы выбрали последнюю стабильную версию (16) из-за ее признанной производительности, высокой

надежности и богатого функционала. С технической точки зрения, PostgreSQL полностью поддерживает стандарт SQL и обеспечивает транзакции ACID (Atomicity, Consistency, Isolation, Durability), что является критически важным требованием при работе с финансовыми данными. Система также обладает развитыми механизмами индексации и репликации, необходимыми для обеспечения быстрого доступа к данным и отказоустойчивости.

Для обеспечения эффективной работы над проектом и взаимодействия с выбранными технологиями используются соответствующие интегрированные среды разработки (IDE) и инструменты.

Основной средой для разработки серверной части на ASP.NET Core является Microsoft Visual Studio 2022. Эта IDE предоставляет разработчикам комплексный набор инструментов, включающий мощный отладчик, интеллектуальное автодополнение кода IntelliSense, глубокую интеграцию с системами контроля версий, такими как Git, инструменты для профилирования производительности приложения, встроенный менеджер пакетов NuGet для удобного управления зависимостями (например, драйвером Npgsql для PostgreSQL или библиотеками для работы с JWT), а также инструменты для непосредственной работы с базами данных.

Для разработки клиентского Android-приложения на Kotlin используется Android Studio (версия Hedgehog | 2023.1.1 или новее). Являясь официальной IDE для Android и построенной на платформе IntelliJ IDEA, она предлагает редактор кода с отличной поддержкой Kotlin и Java, визуальный редактор макетов (Layout Editor), эмулятор Android для тестирования приложения на различных виртуальных устройствах и версиях ОС, разнообразные инструменты для отладки (включая Logcat и Debugger), профилировщик для анализа использования CPU, памяти и сети, интеграцию с системой сборки Gradle и поддержку современных подходов к построению UI, таких как Jetpack Compose.

Для управления базой данных PostgreSQL применяется PgAdmin 4 – популярный инструмент администрирования и разработки с открытым исходным кодом и удобным графическим интерфейсом. Он позволяет легко подключаться к серверам PostgreSQL, просматривать и редактировать данные, управлять различными объектами базы данных (таблицами, схемами, пользователями, правами доступа), выполнять SQL-запросы в специализированном редакторе и отслеживать активность сервера. Этот инструмент незаменим на этапах проектирования схемы БД, наполнения ее тестовыми данными и анализа хранимой информации в процессе разработки и отладки системы.

Таким образом, выбранная архитектура разрабатываемой информационной системы в сочетании с технологическим стеком, включающим ASP.NET Core 8, Kotlin SDK и PostgreSQL 16, а также с использованием средств разработки Visual Studio 2022, Android Studio и PgAdmin 4, нацелена на создание современного, надежного и эффективного прототипа.

Эти технологии предоставляют все необходимые функциональные возможности для реализации ключевых задач проекта: от обработки транзакций и их анализа с помощью алгоритмов машинного обучения до надежного хранения данных и, что особенно важно в контексте данной дипломной работы, своевременного и наглядного информирования пользователя о потенциальных мошеннических угрозах через интерфейс мобильного приложения.

Реализуемый подход позволяет продемонстрировать улучшенный пользовательский опыт по сравнению с некоторыми существующими решениями, где акцент зачастую смещен на пост-обработку уже совершенных операций или используются менее интерактивные методы уведомления. Интеграция выбранных компонентов обеспечивает прочную основу для построения системы, способной внести вклад в повышение уровня защищенности пользователей услуг мобильного банкинга.

2.3 Требования к ИС

На основе анализа предметной области и существующих аналогов сформулированы следующие требования к разрабатываемой информационной системе (ИС) для определения мошеннических транзакций и уведомления о них пользователя.

2.3.1. Функциональные требования

Система должна предоставлять пользователю возможность симулировать инициирование банковских транзакций через интерфейс мобильного приложения, рисунок 1.

Передача данных о транзакции: Мобильное приложение должно обеспечивать безопасную передачу данных о симулируемой транзакции (сумма, получатель, тип операции и т.д.) на серверный модуль для дальнейшей обработки, рисунки 2 и 4.

Анализ транзакции на мошенничество: Серверный модуль должен взаимодействовать с модулем анализа и определения мошенничества (Anti-Fraud Engine), использующим ML-модель, для оценки риска симулируемой транзакции, рисунок 4.

Получение результатов проверки: Мобильное приложение должно получать от серверного модуля результат проверки транзакции, включая оценку уровня риска, рисунок 2.

Проактивное уведомление пользователя: В случае выявления высокого риска мошенничества, система должна до завершения симулируемой транзакции отправлять пользователю push-уведомление о подозрительной операции.

Визуализация риска: Мобильное приложение должно наглядно отображать пользователю информацию о результатах проверки транзакции, включая уровень риска.

Подтверждение/Отклонение транзакции: Система должна предоставлять пользователю возможность подтвердить или отклонить подозрительную транзакцию через интерфейс мобильного приложения (согласно диаграмме последовательности на Рисунке 4 и диаграмме деятельности на Рисунке 6).

Симуляция блокировки: При отклонении транзакции пользователем, мобильное приложение должно инициировать запрос на симуляцию блокировки операции и, опционально, симуляцию блокировки карты.

Просмотр истории транзакций: Система должна предоставлять пользователю доступ к истории симулированных транзакций с указанием их статуса и результатов проверки на мошенничество.

Аутентификация и авторизация: Система должна обеспечивать базовые механизмы аутентификации пользователя для доступа к функциям мобильного приложения (в рамках прототипа).

2.3.2. Нефункциональные требования

Производительность: Время отклика системы на действия пользователя в мобильном приложении (например, навигация, инициирование транзакции) не должно превышать 1-2 секунд. Время от инициирования симулируемой транзакции до получения пользователем push-уведомления (в случае высокого риска) не должно превышать 5-7 секунд, чтобы обеспечить возможность проактивного вмешательства. Серверная часть должна быть способна обрабатывать запросы от клиентского приложения без существенных задержек при симуляции нагрузки.

Надежность: Система должна обеспечивать стабильную работу мобильного приложения и серверной части. Данные о транзакциях и пользователях (в рамках прототипа – анонимизированные/симулированные) должны надежно храниться в базе данных PostgreSQL, обеспечивая целостность и доступность

Безопасность: Взаимодействие между мобильным приложением и серверным модулем должно осуществляться по защищенному каналу. Должны быть предусмотрены меры по защите API от несанкционированного доступа.

Удобство использования: Интерфейс мобильного приложения должен быть интуитивно понятным, простым в освоении и использовании для целевой аудитории. Уведомления о риске должны быть сформулированы ясно и однозначно, не вызывая паники у пользователя, но подчеркивая серьезность ситуации. Процесс подтверждения или отклонения подозрительной транзакции должен быть максимально простым и быстрым.

2.3.3. Требования к интерфейсу

Графический интерфейс пользователя (GUI): Разработка нативного GUI для платформы Android с использованием Kotlin SDK и современных практик дизайна.

Отображение транзакций: Четкое и структурированное отображение информации о симулируемых транзакциях (сумма, дата, получатель, статус).

Визуализация риска: Использование понятных визуальных элементов (цветовая шкала, иконки, прогресс-бары) для индикации уровня мошеннического риска транзакции.

Элементы управления: Наличие ясно обозначенных кнопок и интерактивных элементов для выполнения действий: инициирования транзакции, подтверждения/отклонения операции, навигации по приложению.

Уведомления: Использование стандартных механизмов push-уведомлений ОС Android для оперативного информирования пользователя. Текст уведомлений должен быть кратким и информативным.

Навигация: Логичная и простая структура навигации в приложении, обеспечивающая быстрый доступ ко всем основным функциям (транзакции, история, обучение, настройки).

2.3.4. Требования к программному обеспечению

Клиентская часть: ОС Android 7.0 (API 24) – Android 14 (API 34).

Серверная часть:

- Операционная система, совместимая с .NET 8 (Windows 11).
- NET 8 SDK и Runtime.
- СУБД PostgreSQL версии 16.
- Веб-сервер Kestrel (поставляется с ASP.NET Core).

Средства разработки:

- Для серверной части: Microsoft Visual Studio 2022.
- Для клиентской части: Android Studio (Hedgehog | 2023.1.1).
- Для управления БД: PgAdmin 4.
- Система контроля версий: Git.

3 Демонстрация рабочего продукта

В данном разделе представлена визуальная демонстрация результатов разработки прототипа мобильного приложения для платформы Android. Эти скриншоты иллюстрируют практическую реализацию концепций, проектных решений и требований, подробно описанных в предыдущих главах работы. Разработанное приложение является воплощением проведенного аналитического исследования и спроектированной архитектуры, наглядно показывая, как теоретические основы и сформулированные требования трансформировались в функциональный пользовательский интерфейс. Основной фокус демонстрации направлен на ключевые аспекты взаимодействия пользователя с системой в контексте предотвращения мошенничества: симуляцию основных банковских операций (добавление карт, переводы, платежи), отображение истории транзакций с визуальной оценкой риска, и, что является центральным элементом данной работы, механизм проактивного, но ненавязчивого уведомления пользователя о потенциально опасных действиях. Последующие изображения детально раскрывают функционал и интерфейс приложения, разработанного с использованием Kotlin для Android, и показывают, как оно решает поставленную задачу улучшения информированности и контроля пользователя над безопасностью своих финансов в мобильном банкинге.

На рисунке 3.1 показан экран «Создание карты». Этот интерфейс предназначен для симуляции добавления пользователем своих банковских карт в приложение. Пользователь может задать произвольное название карты и указать начальный баланс. Данная функция необходима для первоначальной настройки прототипа и создания объектов (карт), с которыми будут проводиться симулируемые транзакции.

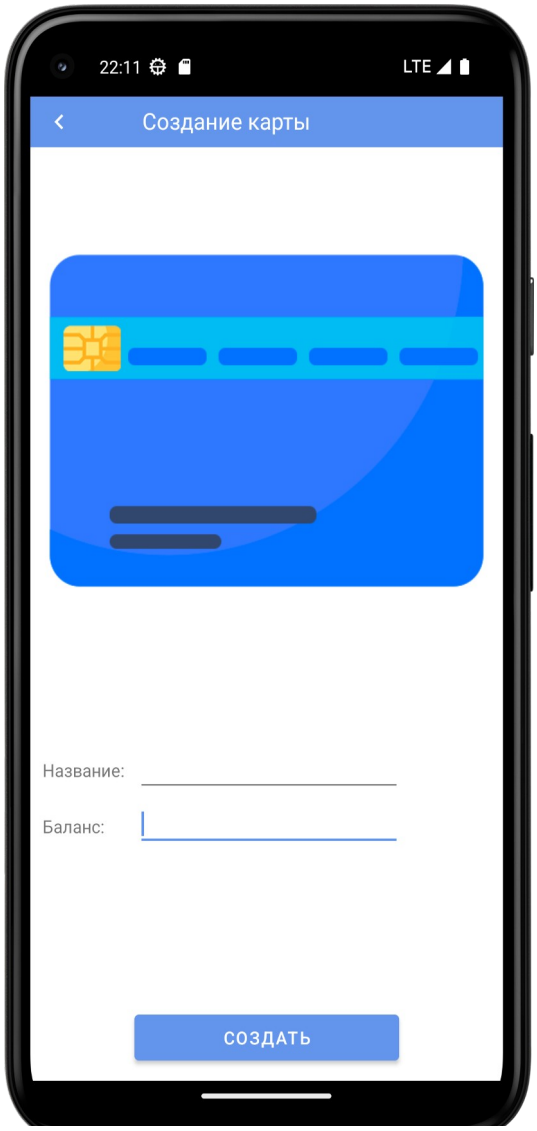

Рисунок 3.1 — Демонстрация мобильного приложения. Создание карты

После добавления карт пользователь видит их список на главном экране вкладки «Карты», как представлено на рисунке 3.2. Здесь отображаются названия карт, их симулированные балансы и последние четыре цифры номера (для имитации реального интерфейса). В правом нижнем углу расположен плавающий элемент управления (Floating Action Button) для быстрого добавления новых карт, то есть для перехода к экрану на рисунке 3.1. Этот экран служит отправной точкой для большинства действий пользователя.

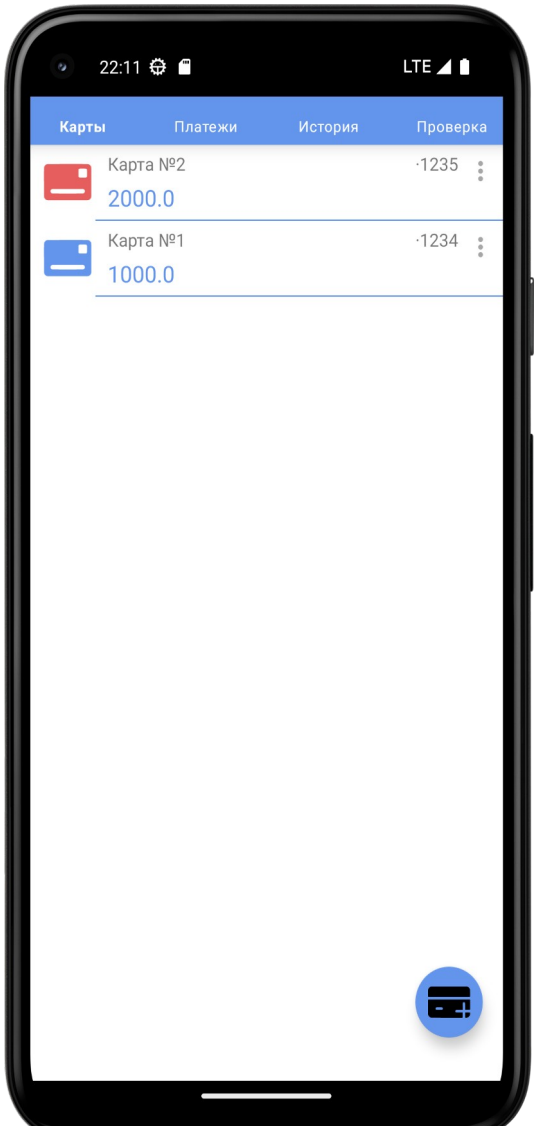

Рисунок 3.2 — Демонстрация мобильного приложения. Вывод списка карт

Перейдя на вкладку «Платежи», рисунок 3.3, пользователь получает доступ к списку доступных типов симулируемых транзакций. В прототипе реализованы основные операции: перевод между своими счетами, перевод на другой счет, оплата мобильной связи и ЖКХ. Выбор одного из этих пунктов инициирует соответствующий сценарий транзакции.

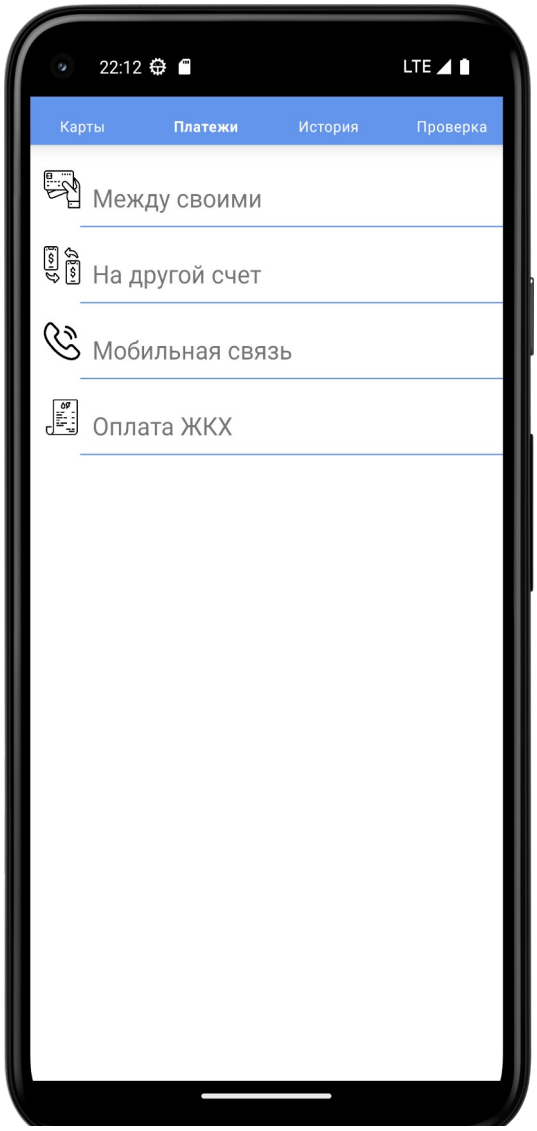

Рисунок 3.3 — Демонстрация мобильного приложения. Вывод типов платежей

Рисунок 3.4 демонстрирует экран процесса совершения транзакции, в данном случае — перевода «Между своими». Пользователю предлагается выбрать карту списания («Откуда») и карту зачисления («Куда») из ранее добавленных на рисунке 3.2, а также ввести сумму перевода («Сколько»). Нажатие кнопки «Оплатить» инициирует отправку данных о транзакции на серверный модуль для анализа риска, согласно бизнес-процессу, описанному в разделе 2.1 и диаграмме на рисунке 2.4.

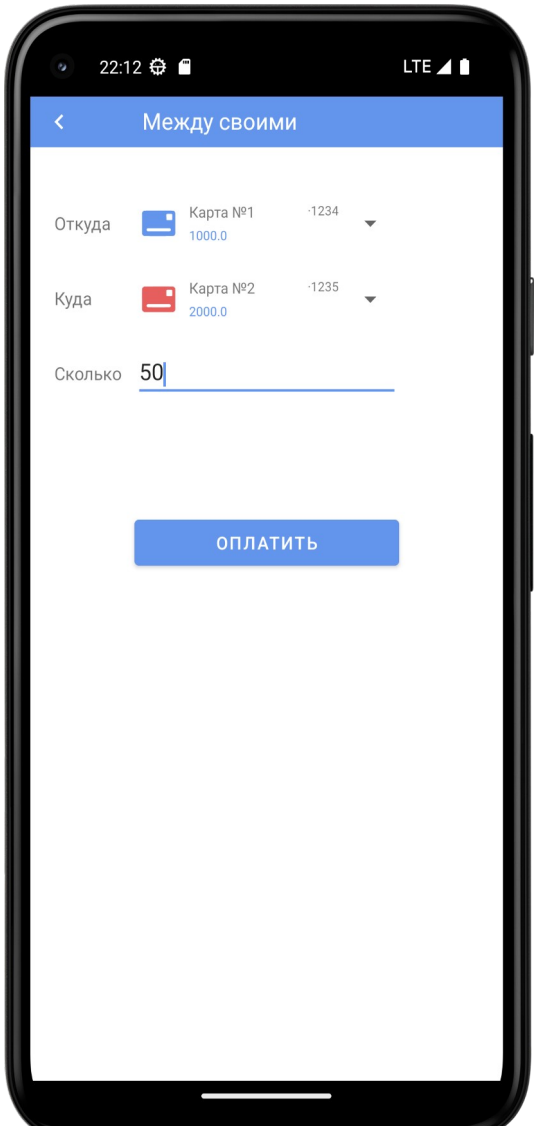

Рисунок 3.4 — Демонстрация мобильного приложения. Процесс совершения транзакции

Результаты совершенных транзакций и их статус проверки на мошенничество отображаются на вкладке «История» изображены на рисунке 3.5. Каждая запись содержит иконку типа платежа, описание, сумму, дату и время. Важным элементом является иконка справа, в виде спидометра с цветовой индикацией, которая визуализирует оценку риска данной транзакции, полученную от антифрод-системы. Это позволяет пользователю ретроспективно оценить безопасность своих операций.

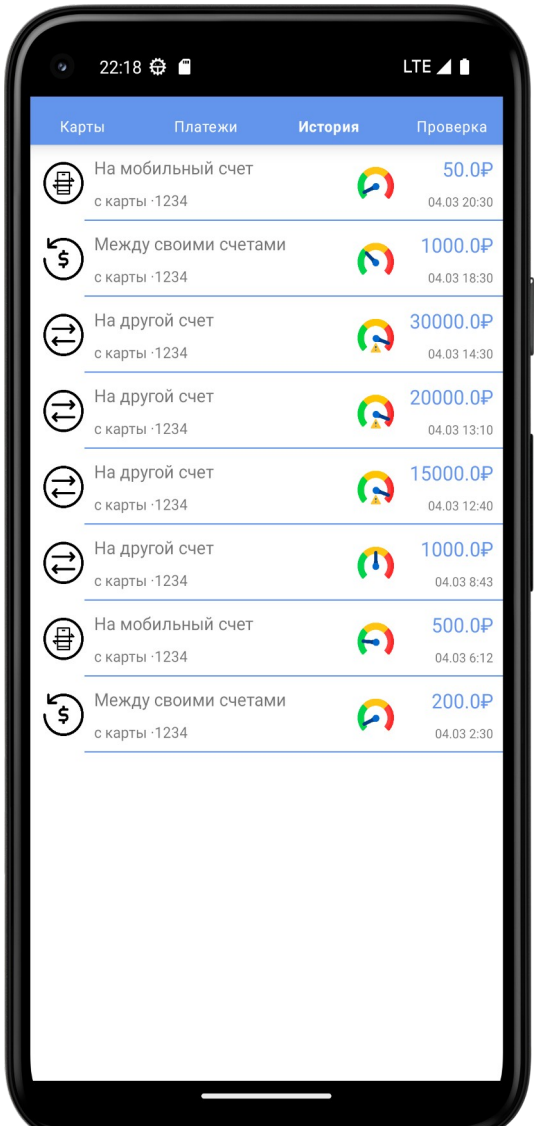

Рисунок 3.5 — Демонстрация мобильного приложения. Вывод истории транзакций

Ключевой особенностью прототипа является механизм проактивного уведомления. Рисунок 3.6 показывает, как система информирует пользователя о высоком риске непосредственно после попытки совершения подозрительной транзакции. Всплывающее сообщение с текстом «Риск мошеннической транзакции!» появляется в нижней части экрана, привлекая внимание пользователя к потенциальной угрозе и позволяя ему предпринять действия до фактического завершения опасной операции.

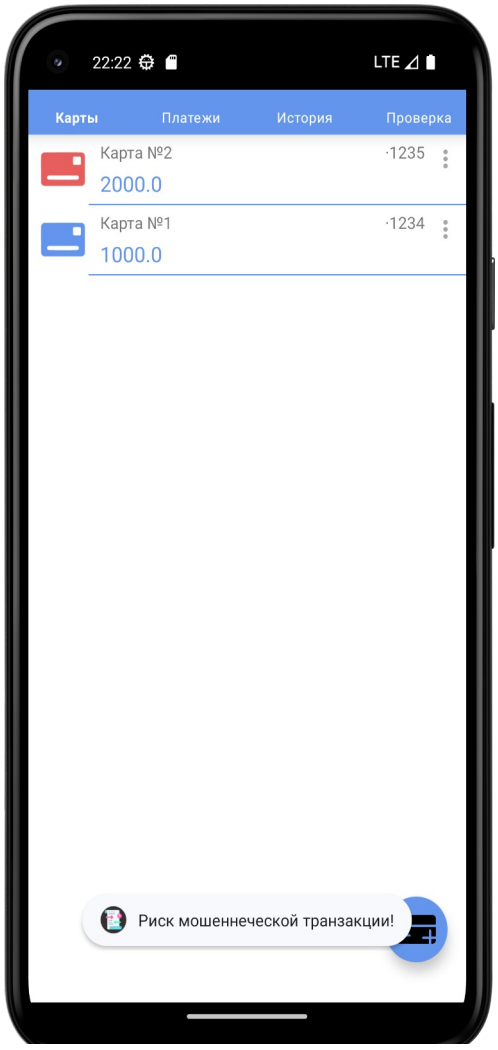

Рисунок 3.6 — Демонстрация мобильного приложения. Уведомление об совершении подозрительной транзакции

Для предоставления пользователю обобщенной картины уровня безопасности его операций в приложении предусмотрена вкладка «Проверка», представленная на рисунке 3.7. Здесь представлена визуализация данных о рисках транзакций за определенный период в виде графика. Ось X отображает время, ось Y – процентный уровень риска. Цветовой градиент области под графиком (от зеленого к красному) интуитивно показывает периоды с низким и высоким уровнем подозрительной активности. Это дает пользователю дополнительный инструмент для анализа безопасности своих финансов.

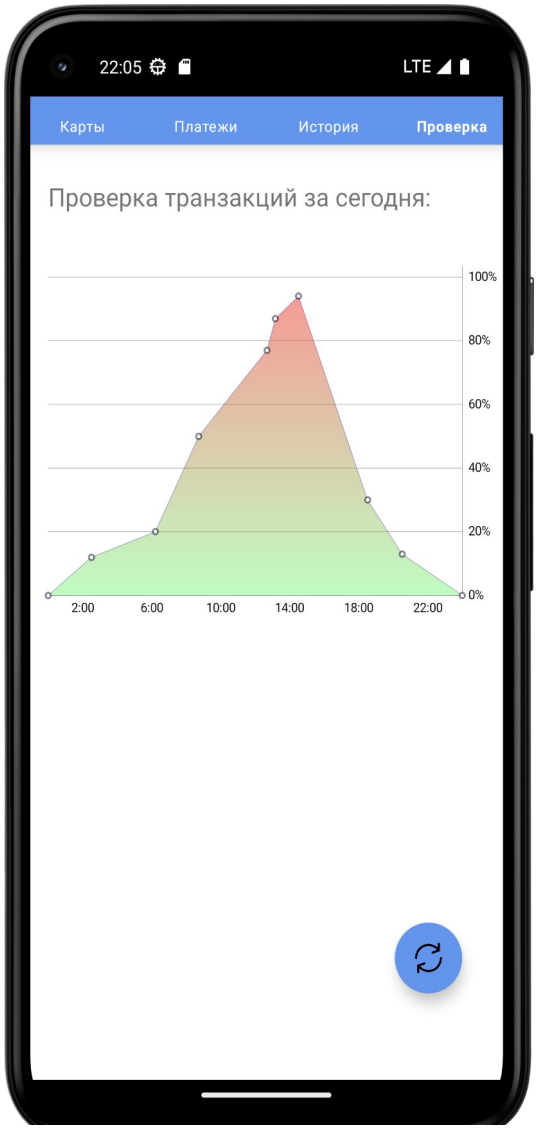

Рисунок 3.7 — Демонстрация мобильного приложения. Визуализация уровня риска транзакций

Представленные экраны демонстрируют реализацию основных функциональных требований к прототипу, включая симуляцию банковских операций, взаимодействие с системой оценки рисков, наглядное отображение результатов проверки и, что наиболее важно, механизм проактивного уведомления пользователя о потенциально мошеннических транзакциях, а также средства для визуального анализа уровня риска. Пользовательский интерфейс спроектирован с учетом требований удобства использования.

ЗАКЛЮЧЕНИЕ

В ходе прохождения практики были выполнены следующие ключевые этапы, заложившие основу для дальнейшей разработки:

- Проведен глубокий анализ предметной области: Изучены современные угрозы мобильного банковского мошенничества (фишинг, социальная инженерия, вредоносные ПО, DeepFake-технологии), а также существующие методы противодействия.
- Исследованы аналоги: Проанализированы антифрод-решения и подходы к уведомлению пользователей в ведущих российских банках (СберБанк, ВТБ, Т-Банк) и регуляторные инициативы (ЦБ РФ), выявлены их сильные стороны и ограничения, что позволило обосновать актуальность предлагаемого подхода.
- Выполнено проектирование информационной системы: Разработаны модели бизнес-процессов взаимодействия пользователя с системой с использованием нотации UML (диаграммы вариантов использования, последовательности, деятельности), наглядно демонстрирующие предлагаемый сценарий проактивного информирования.
- Определена сервис-ориентированная архитектура системы, включающая клиентский модуль (мобильное приложение), серверный модуль (Backend), модуль анализа и базу данных.
- Подобран современный технологический стек (ASP.NET Core 8, Kotlin SDK, PostgreSQL 16) и средства разработки (Visual Studio 2022, Android Studio, PgAdmin 4), отвечающие задачам проекта.
- Сформулированы детальные функциональные, нефункциональные требования, а также требования к интерфейсу и программно-аппаратному обеспечению.
- Разработан и продемонстрирован протопит мобильного приложения

Основным результатом преддипломной практики является комплексное проектирование информационной системы и создание теоретической и технологической базы для разработки прототипа. Ключевое отличие проектируемого решения заключается в акценте на проактивное уведомление пользователя о высоком риске до завершения транзакции, предоставлении наглядной визуализации риска и интуитивно понятных инструментов для подтверждения или отклонения операции, а также интерактивном обучении, что направлено на повышение безопасности и улучшение пользовательского опыта по сравнению с существующими подходами.

Задачи, поставленные перед преддипломной практикой, были успешно выполнены. Приобретены и закреплены навыки анализа предметной области, системного проектирования, моделирования бизнес-процессов, выбора современных технологий и инструментов разработки. Создана необходимая основа для завершения разработки прототипа мобильного приложения в рамках выпускной квалификационной работы. Следующими шагами станут полная реализация функционала серверной и клиентской частей, их тестирование и описание полученных результатов в итоговой работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Группа IB. Тренды мошенничества в 2024–2025 гг. URL: https://www.group-ib.ru (дата обращения: 02.04.2025)

2. Положительные технологии. Уязвимости мобильных банкингов – 2025. URL: https://www.ptsecurity.com (дата обращения: 02.04.2025)

3. Отчет Национальной службы кибербезопасности. Годовой отчет – 2024. URL: https://nbki.ru (дата обращения: 02.04.2025)

4. Центральный банк Российской Федерации. Указание №5434-У «О стандартах биометрической аутентификации» (2025 г.). URL: https://cbr.ru (дата обращения: 02.04.2025)

5. Лаборатория Касперского. Социальная инженерия в РФ – 2025. URL: https://www.kaspersky.ru (дата обращения: 02.04.2025)

6. Минцифра России. Постановление Правительства РФ №235-Пр «О мерах защиты пользователей электронных платёжных систем». URL: https://digital.gov.ru (дата обращения: 02.04.2025)

7. Банк России. Отчет «О состоянии рынка платежных услуг за 2023 год». URL: https://cbr.ru (дата обращения: 19.04.2025)

8. СберБанк. Техническая документация API «Sberbank AI Solutions». URL: https://developer.sber.ru (дата обращения: 19.04.2025)

9. ВТБ. Пресс-релиз «Запуск системы VTB Anti-Fraud». URL: https://vtb.ru/press (дата обращения: 19.04.2025)

10. Петров А.И. Интервью с СТО Т-Банка // Журнал «Банковские технологии». – 2024. – №3. – С. 45–49.

11. Европейский Союз. Директива 2024/0078 «О платежных услугах (PSD3)». URL: https://eur-lex.europa.eu (дата обращения: 19.04.2025).

12. Тинькофф Банк. Отчет «Итоги внедрения динамической биометрии». URL: https://www.tinkoff.ru (дата обращения: 21.04.2025).

13. НИУ ВШЭ. Исследование «Поведенческие паттерны пользователей мобильного банкинга». – М.: Изд-во ВШЭ, 2024. – 134 с.

14. Нотация и семантика языка UML: Информация Автор: Александр Леоненков | Школа IT-менеджмента АНХ при Правительстве РФ http://www.intuit.ru/studies/courses/32/32/info (дата обращения: 22.04.2025)

15. Язык UML 2 в анализе и проектировании программных систем и бизнес-процессов: Информация Автор: Александр Леоненков http://www.intuit.ru/studies/courses/480/336/info (дата обращения: 22.04.2025)

16. Хомоненко А.Д., Басыров А.Г., Бубнов В.П., Забродин А.В., Краснов С.А., Лохвицкий В.А.,Тырва А.В. Модели и методы исследования информационных систем: монография Издательство "Лань" 204.с. - 2019г https://e.lanbook.com/reader/book/119640#94 (дата обращения: 22.04.2025)

17. Моделирование бизнес-процессов: учебное пособие Кравченко А. В.,Драгунова Е. В.,Кириллов Ю. В. Издательство Лань Новосибирский государственный технический университет , 136 стр., 2020г. https://e.lanbook.com/book/152364 (дата обращения: 22.04.2025)