**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 работе 29 страниц, 7 рисунков, 2 таблицы, 5 источников.

Ключевые слова: *рельсошлифавальный поезд, ремонтные поперечные профили, десктопное приложение, геометрические параметры, железнодорожные пути.*

Предметная область – десктопное приложение «Расчет» предназначено для использования на рельсошлифовальных поездах на этапе подготовки к работам и проектирования технологического процесса шлифования рельсов. Основная функция программы заключается в расчете геометрических параметров ремонтных поперечных профилей рельсов. Приложение предназначено для эксплуатации как во время планово-подготовительных работ на перегоне, так и при наличии данных без выезда на перегон. Разработка данного программного продукта направлена на повышение эффективности процесса обслуживания и ремонта железнодорожных путей.

**ABSTRACT**

The work contains 29 pages, 7 figures, 2 tables, 5 sources.

Keywords: *a rail grinding train, a repair cross profiles, a desktop application, geometric parameters, a railway tracks.*

The desktop application «Calculation» is designed for use on rail grinding trains during preparation for work and planning the rail grinding process. The main function of the software is to calculate the geometric parameters of repair cross profiles of rails. The program is intended to be used both during planned preparatory works on the track and, when data is available, without going out onto the track. Development of this software aims to enhance the efficiency of railway track maintenance and repair processes.

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1. Visual Studio – среда разработки от компании Microsoft.

# ПЕРЕЧЕНЬ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

1. ИС – информационная система.
2. .NET – программная платформа компании Microsoft.
3. MS – компания Microsoft.
4. IDE – интегрированная среда разработки.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#__RefHeading___Toc2410_2887410827)

[1 Аналитическое исследование 7](#__RefHeading___Toc6444_3885207264)

[1.1 Описание предметной области 7](#__RefHeading___Toc6446_3885207264)

[1.2 Анализ аналогов 9](#__RefHeading___Toc6502_3885207264)

[1.3 Актуальность разработки 11](#__RefHeading___Toc6504_3885207264)

[2 Проектирование информационной системы 13](#__RefHeading___Toc6448_3885207264)

[2.1 Моделирование бизнес-процессов ИС 13](#__RefHeading___Toc7440_3885207264)

[2.2 Структура ИС и ее средства разработки 16](#__RefHeading___Toc7442_3885207264)

[2.3 Требования к ИС 22](#__RefHeading___Toc7444_3885207264)

[3 Разработка информационной системы 27](#__RefHeading___Toc6450_3885207264)

[3.1 Описание структуры базы данных 27](#__RefHeading___Toc6452_3885207264)

[3.2 Разработка серверной части приложения 27](#__RefHeading___Toc6454_3885207264)

[3.3 Разработка клиентской части приложения 27](#__RefHeading___Toc6454_3885207264_%25D0%)

[3.4 Интеграция с сервисом по определению мошеннических транзакций 27](#__RefHeading___Toc6456_3885207264)

[ЗАКЛЮЧЕНИЕ 28](#__RefHeading___Toc2420_2887410827)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#__RefHeading___Toc2422_2887410827)

# ВВЕДЕНИЕ

* 1.1. Актуальность темы
* 1.2. Цели и задачи работы
* 1.3. Объект и предмет исследования

В рамках выпускной квалификационной работы рассматривается разработка мобильных программных продуктов и ее актуальность на текущий момент времени. Рассматривается предметная область доставки посылок и проблема организации частной доставки. Производится анализ существующих решений в виде мобильных приложений на платформе Android. Целью работы является разработка собственного на платформе Androidприложения для организации частной доставки, учитывающее недостатки существующих решений. Для этого рассматриваются инструменты создания мобильных приложений. Производится подбор оптимальных технологических решений для решения поставленной задачи. Вслед за этапами проектирования и выбора технологических решений начинается разработка Android-приложения на языке программирования Java. При разработке используется база данных класса NoSQL Firebase Realtime Database и дополнительные сервисы Firebase. По завершении этапа разработки, производится альфа-тестирование разработанного приложения при помощи: smoke-тестирования, UAT-тестирования и функционального тестирования. Результатом работы является Android-приложение, отличающееся от представленных ранее удобством использования и направленностью на потребности частных лиц. Приложение получило название «Pick And Go».

## ****Аналитическое исследование****

### 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.

### 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].

### 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.

## ****Проектирование информационной системы****

### 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» и «Заблокировать карту». Эти варианты использования были соединены с актором связью «Association», что позволяет отразить взаимодействие пользователя с системой (Рисунок 1).

Пользователь:

Сценарии:

«Совершение транзакции»

«Заблокировать карту».

Рисунок 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) [15].

Мобильное приложение:

Сценарии:

«Отправить данные о транзакции»

«Получить данные о проверке транзакции»

«Отобразить пользователю результаты проверки транзакции»

«Отобразить пользователю push-уведомление»

«Послать запрос на блокировку карты»

Рисунок 2 — Добавление сценариев для мобильного приложения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Для построения диаграммы последовательности, описывающей процесс проверки транзакции, на диаграмму был помещен актор — пользователь. Также были добавлены линии жизни объектов, таких как мобильное приложение, банк и антифрод-система (Рисунок 3).

Рисунок 3 — Добавление акторов и линий жизни объектов

Далее были добавлены последовательные сообщения между актором и объектами. Антифрод-система отправляет запрос на получение истории транзакций в базу данных, после чего база данных возвращает данные. Затем антифрод-система обращается к ML-модели для оценки риска, которая возвращает результат. В случае выявления высокого риска, антифрод-система отправляет команду на блокировку в мобильное приложение, которое уведомляет пользователя о подозрительной операции и предлагает варианты «Подтвердить» или «Отклонить» (Рисунок 4) [16].

Sequence Diagram:

(Пользователь, мобильное приложение, Антифрод-система, банк)

Антифрод-система → База данных: Запрос истории транзакций.

База данных → Антифрод-система: Возврат данных.

Антифрод-система → ML-модель: Оценка риска.

ML-модель → Антифрод-система: Результат (высокий риск).

Антифрод-система → Мобильное приложение: Команда на блокировку.

Банк → Мобильное приложение: Отправка уведомления о подозрительной операции.

Мобильное приложение → Пользователь: Уведомление.

Мобильное приложение → Пользователь: Отображение опций «Подтвердить»/«Отклонить».

Пользователь → Мобильное приложение: Выбор «Отклонить».

Мобильное приложение → Банк: Запрос на блокировку транзакции.

Банк → Мобильное приложение: Подтверждение блокировки.

Рисунок 4 — Добавление последовательных сообщений

Для построения диаграммы деятельности, описывающей процесс обработки уведомления о подозрительной транзакции, на диаграмму были помещены действия «Получить уведомление» и «Заблокировать карту» (Рисунок 5).

Рисунок 5 — Добавление действий

Далее была добавлена логика принятия решения: если пользователь отклоняет операцию, система выполняет действие «Заблокировать карту» и уведомляет банк. В противном случае, если пользователь подтверждает операцию, процесс завершается (Рисунок 6) [17].

Activity Diagram:

[Старт] → Получить уведомление →

Если (пользователь отклоняет операцию):

→ Заблокировать карту → Уведомить банк → [Конец].

Иначе:

→ Подтвердить операцию → [Конец].

Рисунок 6 — Логика принятия решения в диаграмме деятельности

Предложенные бизнес-процессы учитывают опыт СберБанка и ВТБ. В отличие от аналогов, акцент сделан на использовании ML для оценки риска до завершения транзакции и мгновенном уведомление пользователя, а не постфактумная блокировка.

### Структура ИС и ее средства разработки

Разрабатываемое в рамках данной дипломной работы программное обеспечение представляет собой прототип информационной системы, нацеленной на демонстрацию улучшенных механизмов взаимодействия с пользователем в контексте предотвращения мошеннических транзакций в мобильных банковских приложениях. Основная цель – не просто блокировать подозрительные операции постфактум, как это часто реализуется в существующих системах (см. раздел 1.2), а проактивно информировать пользователя о потенциальных рисках до завершения транзакции, предоставляя ему наглядную информацию и возможность осознанного принятия решения. Ключевое функциональное назначение данного ПО заключается в визуализации результатов работы антифрод-системы в удобном и интуитивно понятном для конечного пользователя виде, а также в демонстрации процесса анализа транзакций с использованием алгоритмов машинного обучения для выявления аномалий и подозрительной активности.

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.

### 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.

### 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).

### Нефункциональные требования

Нефункциональные требования описывают атрибуты качества системы и ограничения на ее работу.

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.

### Требования к интерфейсу

Требования к интерфейсу детализируют взаимодействие пользователя с системой.

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).

### 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.

## ****Разработка информационной системы****

### Описание структуры базы данных

Привести диаграмму классов с описанием таблиц в ней.

### Разработка серверной части приложения

Описать разработку backend приложения на Asp.net

### Разработка клиентской части приложения

Описать разработку мобильного приложения на Kotlin

### Интеграция с сервисом по определению мошеннических транзакций

Описать интеграцию системы с сервисом по определению мошеннических транзакций на Python.

# ЗАКЛЮЧЕНИЕ

Прохождение технологической практики представляет собой один из важнейших этапов в учебном процессе, поскольку оно позволяет студентам приобрести опыт работы в организации, применять теоретические знания на практике, поднимать уровень своей профессиональной подготовки и углублять свое понимание особенностей профессиональной деятельности.

Задачи, поставленные перед технологической практикой, были успешно выполнены, и цель получения профессиональных навыков и опыта в профессиональной сфере, в частности, в разработке программных продуктов, необходимых для внутренних потребностей компании, была достигнута.

Одним из результатов прохождения производственной практики является создание сценариев диалога с общим чат-ботом по внутреннем портале ВиКо по вопросам работы с: анализом профессиональных рисков, ответов на запросы к руководителям, оценке и ведении данных о нормах средств индивидуальной защиты.

# 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 CTO 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)