МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**КУРСОВОЙ ПРОЕКТ**

по дисциплине: «Компьютерные системы и сети**»**

на тему: «Сетевое ПО «Банковское приложение»»

Выполнил**:** ст. гр. 10701323 Шаплавский Н.С.

Приняла**:** ст. пр. Белова С.В.

Минск 2025

**ОГЛАВЛЕНИЕ**

**Элементы оглавления не найдены.**

# ВВЕДЕНИЕ

В современных условиях стремительного развития информационных технологий эффективное управление банковскими сервисами невозможно без внедрения специализированного программного обеспечения. Приложение банкинг как ключевой элемент финансовых услуг, требует автоматизации процессов обработки платежей, учета клиентов, управления транзакциями, контроля доступа к данным и обеспечения информационной безопасности.

Банковские приложения становятся важнейшим инструментом для оптимизации работы финансовых учреждений, повышения удобства пользователей и конкурентоспособности банка в целом. Целью данного курсового проекта является разработка концепции банковского приложения, адаптированного под потребности клиентов, с учетом особенностей его инфраструктуры и бизнес-процессов.

В работе рассматриваются вопросы интеграции платежных систем, модулей управления счетами, инструментов мониторинга транзакционной активности и защиты данных. Особое внимание уделяется проектированию архитектуры, выбору технологий и протоколов, обеспечивающих стабильность, масштабируемость и безопасность решения.

Курсовой проект актуален, потому что сегодня практически все финансовые операции переходят в цифровое пространство, и банки — не исключение. Современным финансовым учреждениям необходимо не только оперативно обрабатывать платежи и переводы, но и надёжно хранить данные клиентов, контролировать доступ к счетам, а также предотвращать ошибки и мошенничество.

1. **Обзор состояния вопроса**

### 1.1. Описание и анализ предметной области

Банковское программное обеспечение предназначено для упрощения и автоматизации банковских операций, таких как управление счетами, переводы, оплата услуг и мониторинг финансового состояния пользователя. При авторизации клиент вводит свои учетные данные, которые проверяются через систему безопасности. После успешной идентификации пользователю предоставляется доступ к его финансовой информации, включая баланс и доступные услуги.

Каждая операция в системе фиксируется в базе данных, обеспечивая безопасность и защиту от несанкционированного доступа. Пользователь может проводить финансовые операции, такие как переводы между своими счетами а так же другим пользователям. Все транзакции проходят через систему верификации и шифруются для защиты персональных данных.

В системе предусмотрены разграничения прав доступа: обычные пользователи могут управлять своими счетами и картами, сотрудники банка могут просматривать клиентские заявки и выполнять операции согласно своим полномочиям, а администраторы управляют пользователями и настройками безопасности.

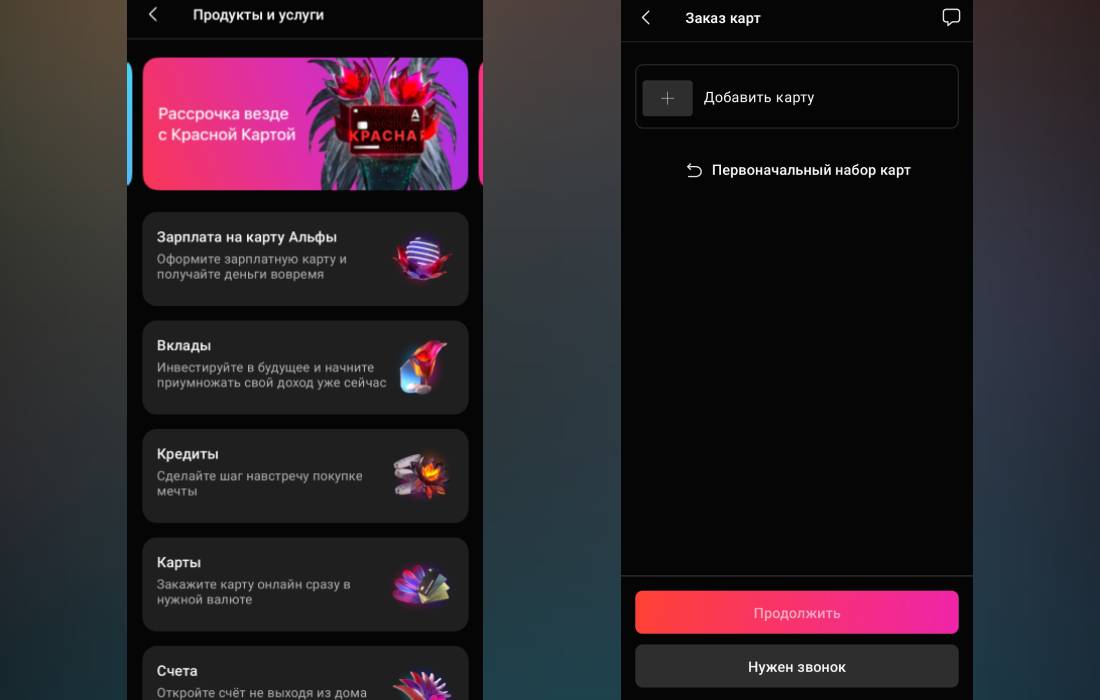
Разработка мобильного банкинга в рамках курсового проекта фокусируется на базовых функциях, таких как управление балансом, переводы, защита данных и удобный интерфейс. Для реализации используется **графический интерфейс на Qt Design**, серверная часть написана на **Python с использованием socket**, а база данных **PostgreSQL** обеспечивает надежное хранение информации. Для безопасной работы применяются **шифрование данных** и **протоколы аутентификации**, что гарантирует стабильность и защиту пользовательских данных.

### 1.2. Обзор аналогов

В ходе исследования были рассмотрены три популярных мобильных банковских приложения: **Айсинк (Альфа-Банк Беларусь), Беларусьбанк, Тинькофф**.

#### **Айсинк (Альфа-Банк Беларусь)**

Приложение от Альфа-Банка Беларусь с современным и удобным интерфейсом. Отличается возможностью **открытия счетов онлайн, управления картами и кредитами, оплаты коммунальных услуг и перевода денег между банками**. В Айсинк встроен **чат-бот**, который помогает клиентам решать вопросы без необходимости звонить в поддержку. Также есть система бонусов и кэшбэка. Однако приложение доступно только для клиентов Альфа-Банка и не поддерживает мультивалютные счета.

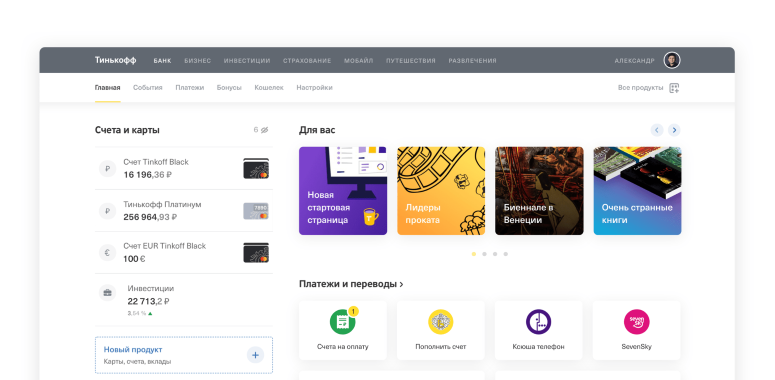


#### **Беларусьбанк**

Официальное приложение крупнейшего государственного банка Беларуси. Оно предоставляет **доступ к депозитам, кредитам, переводам, оплате услуг и обмену валют**. Важным преимуществом является **широкая сеть банкоматов и отделений**, а также интеграция с **ЕРИП (единая система платежей в Беларуси)**. Однако пользователи отмечают, что **интерфейс приложения менее интуитивен**, чем у частных банков, и иногда возникают технические сбои.



#### **Тинькофф**

Одно из самых продвинутых банковских приложений, которое работает полностью в цифровом формате. В нем есть **гибкие настройки карт, инвестиции, накопительные счета, управление подписками, кредиты и страховки**. Приложение **автоматически анализирует расходы**, предлагает удобные финансовые категории и имеет **интеграцию с Apple Pay и Google Pay**. Однако оно ориентировано на российский рынок и не поддерживает все функции за пределами России.

* 1. **Изучение используемых технологий и алгоритмов**

При разработке клиент-серверного приложения «Банкинг» основной упор сделан на сокетное взаимодействие между клиентом и сервером, реализованным на языке Python. Это обеспечивает прямое двустороннее соединение и позволяет передавать данные в режиме реального времени без использования HTTP-протокола.

Серверная часть будет реализована с использованием стандартных средств Python (модуля socket) и поддерживать многопоточную архитектуру. Каждый клиент подключается к серверу через отдельный поток или процесс, что обеспечивает параллельную обработку запросов и предотвращает блокировку основного цикла приложения.

Для хранения данных используется СУБД PostgreSQL. Взаимодействие с базой данных организовано через ORM-библиотеки (SQLAlchemy), что упрощает работу с таблицами и автоматизирует выполнение SQL-запросов, а так же делает позволяет сделать транзакции атомарными. В структуру базы данных входят таблицы, отражающие основные сущности приложения: пользователи, банковские счета, транзакции, заявки и т.п.

Клиентское приложение также написано на Python, и при запуске оно устанавливает сокет-соединение с сервером. Для обеспечения отзывчивости интерфейса и избежания блокировки пользовательского интерфейса сетевое взаимодействие выносится в отдельные потоки.

Обмен данными между клиентом и сервером осуществляется в формате JSON, что делает структуру сообщений удобной для парсинга и расширяемой в будущем.

Такой подход обеспечивает гибкость, масштабируемость и прямой контроль над сетевым взаимодействием, что особенно важно для приложений в банковской сфере, где критически важны скорость и надежность передачи данных.

## 2. Разработка требований к программному обеспечению

Целью настоящего курсового проекта является создание удобной, надёжной и масштабируемой банковской системы, реализованной с использованием технологий Python, сокетного взаимодействия и СУБД PostgreSQL. Такая система должна автоматизировать ключевые финансовые процессы, повысить безопасность хранения и передачи данных, а также обеспечить эффективное взаимодействие между клиентами и сотрудниками банка.

Основными задачами разработки являются:

* обеспечение безопасного и стабильного обмена данными в режиме реального времени между клиентом и сервером;
* организация учёта клиентов, счетов и транзакций;
* реализация гибкой модели прав доступа;
* минимизация человеческого фактора и ускорение обработки запросов.

На основании этих целей формируются функциональные и нефункциональные требования.

2.1 Ролевое распределение

Приложение предусматривает три основные роли:

* **Оператор** (администратор) — ведет контроль за стабильной работой приложения, имеет полный доступ к бд, для исправления ошибок, а так же проведеня расследований
* **Сотрудник** — имеет ограниченный доступ: просмотр чужих профилей, контроль баланса и истории действий.

#### Роль Оператор:

1. Управление клиентами:
   * Регистрация: ФИО, email, паспортные данные.
2. Управление банковскими счетами:
   * Создание новых счетов для клиентов.
   * Просмотр баланса и операций по счёту.
   * Изменение статуса счета: активен, заблокирован, закрыт.
3. Работа с транзакциями:
   * Создание и проведение перевода между счетами.
   * Просмотр истории операций.
4. Автоматизация:
   * Автоматическое изменение статуса счёта после завершения действия (например, после перевода).
   * Ведение журнала всех операций.

#### Роль Сотрудник:

1. Просмотр собственного профиля:
   * Информация: ФИО, должность, зарплата, рабочие часы.
   * Доступ к чужой истории активности (например, история транзакций).

### 2.3 Нефункциональные требования

#### 1. Производительность и надёжность:

* Время отклика при передаче данных не должно превышать 2 секунд.
* Система должна работать стабильно 24/7.

#### 2. Безопасность:

* Разграничение прав доступа: сотрудники не могут изменять свои данные; менеджеры не имеют доступа к счетам клиентов.
* Хранение паролей и личных данных в хешированном виде.
* Шифрование всех передаваемых по сокетам данных.

#### 3. Удобство использования:

* Простой и понятный интерфейс
* Основные разделы: сотрудники, клиенты, счета, транзакции.
* Четкие названия всех кнопок и действий, соответствие логике бизнес-процессов.

#### 4. Используемые технологии:

* Клиентская часть: Python с GUI (CTK).
* Серверная часть: Python + socket, многопоточность
* База данных: PostgreSQL.
* Формат передачи данных: JSON.
* Протокол: TCP/IP.