### Частное учреждение образования

### Колледж бизнеса и права

### СОЗДАНИЕ ПРИЛОЖЕНИЯ ПО АВТОМАТИЗАЦИИ РАБОТЫ БАНКА «Indusland Bank»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### к курсовому проектированию по дисциплине

### «Конструирование программ и языки программирования»

КП Т.196001.401 ПЗ

Руководитель проекта (А.В.Кривошеина)

Учащийся (Н. М. Абдусаломов)

Минск, 2024

**СОДЕРЖАНИЕ**

Изм.

Изм.

Лист

Лист

№ докум.

№ докум.

Подпись

Подпись

Дата

Дата

Лист

Лист

3

3

КП Т.196001.401 ПЗ

ПЗ

ДП Т.517074.401 ПЗ

Разраб.

Разраб.

Абдусаломов Н. М.

Мовчан П.И.

Провер.

Провер.

Кривошеина А.В.

Багласова Е.В.

Т. контр.

Т. контр.

*.*

*Якимович К.О.*

Н. контр.

Н. контр.

Багласова Е.В.

Утверд.

Утверд.

*Багласова Т.Г.*

*Программное средство для автоматизации банковских операций*

*Программное средство для автоматизации учета заявок жителей Фрунзенского района г.Минска*

Лит.

Лит.

Листов

Листов

54

54

КБП

КБП

[Введение 3](#_Toc168502529)

[1. Описание задачи 4](#_Toc168502530)

[1.1. Анализ предметной области 4](#_Toc168502531)

[1.2. Постановка задачи 5](#_Toc168502532)

[2. Проектирование системы 7](#_Toc168502533)

[2.1. Требования к приложению 7](#_Toc168502534)

[2.2. Проектирование модели 7](#_Toc168502535)

[2.3. Организация данных 11](#_Toc168502536)

[2.4. Концептуальный прототип 13](#_Toc168502537)

[3. Описание реализации программного средства 27](#_Toc168502538)

[3.1. Инструменты разработки и применения технологии 27](#_Toc168502539)

[3.2. Порядок авторизации пользователей 27](#_Toc168502540)

[3.3. Организация данных 28](#_Toc168502541)

[3.4. Функции: обоснование выбранной организации 30](#_Toc168502542)

[3.5. Входные и выходные данные 32](#_Toc168502543)

[3.6. Функциональное тестирование 32](#_Toc168502544)

[3.7. Описание справочной системы 44](#_Toc168502545)

[4. Применение 46](#_Toc168502546)

[4.1. Назначение программного средства 46](#_Toc168502547)

[4.2. Условия применения 46](#_Toc168502548)

[Заключение 48](#_Toc168502549)

[Список использованных источников 49](#_Toc168502550)

[Приложение А Текст программных модулей 50](#_Toc168502551)

[Приложение Б Выходные документы 54](#_Toc168502552)

# Введение

В современном мире, где технологии проникают во все сферы жизни, банковская индустрия не является исключением. Автоматизация процессов становится все более актуальной и важной задачей, позволяющей минимизировать потери рабочего времени сотрудников, экономить на численности персонала и повышать уровень обслуживания клиентов.

Целью курсового проекта «Автоматизация банковских операций - банкомат» является создание программного обеспечения, которое позволит автоматизировать работу банкоматов, улучшая их функциональность и надежность. Это программное средство должно обеспечивать удобство для клиентов при выполнении операций, таких как снятие наличных, проверка баланса, переводы и оплата счетов, а также повышать безопасность и эффективность работы банковских учреждений.

Пояснительная записка к курсовому проекту содержит пять разделов:

Описание задачи: В этом разделе раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть решены. Описываются существующие аналоги.

Проектирование системы: В этом разделе перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения. Описываются требования к приложению, строится концептуальный прототип, описывается логическая и физическая организация данных, проектируется справочная система.

Описание реализации программного средства: В этом разделе представлены общие сведения о программном средстве и его функциональном назначении, входные и выходные данные.

Применение: Этот раздел предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы.

Заключение: В заключении будет проанализировано созданное программное средство, определена степень соответствия поставленной задачи и выполненной работы.

Приложение А будет содержать текст программы.

Графическая часть содержит в себе все необходимые диаграммы для проектирования данного программного средства, такие как:

* диаграмма вариантов использования;
* диаграмма деятельности;
* диаграмма классов;
* диаграмма последовательности.

# Описание задачи

# Анализ предметной области

В современном мире банковское дело играет ключевую роль в экономической системе каждого государства. Банковский сектор обеспечивает стабильность и эффективность финансовых операций, а также обеспечивает доступ к финансовым услугам для всех слоев населения. В этом контексте, банкоматы выступают важным звеном в сети банковского обслуживания, предоставляя клиентам возможность проводить финансовые операции независимо от местоположения и времени суток.

Банкоматы представляют собой важное звено в банковском обслуживании, предоставляя клиентам широкий спектр возможностей для управления своими финансами. Далее перечислено описание основных услуг, которые предоставляются через банкоматы:

* Выдача наличных – это одна из основных функций банкомата. Клиенты могут получить наличные деньги с своего банковского счета, используя свою банковскую карту и соответствующий ПИН-код. Эта услуга особенно важна для тех ситуаций, когда требуется наличность, например, для покупки в местах, где не принимают пластиковые карты.
* Пополнение счета – банкоматы также позволяют клиентам вносить наличные или чеки на свой банковский счет. Это удобно, когда нет доступа к филиалу банка или кассовому отделу.
* Оплата счетов – банкоматы предоставляют возможность оплаты различных счетов, таких как налоговые платежи, оплата за обучение, энергоснабжение, ЖКХ (жилищно-коммунальные услуги) и другие коммунальные услуги. Это позволяет клиентам эффективно управлять своими финансами и избегать просрочек платежей.
* Оплата штрафов – банкоматы также предоставляют функцию оплаты штрафов и других сборов. Это удобно для быстрого урегулирования обязательных платежей.
* Переводы средств – клиенты могут осуществлять переводы денег между своими счетами или на счета других клиентов того же банка или даже других банков.

Банковская автоматизация охватывает широкий спектр задач, направленных на улучшение и оптимизацию операций с банкоматами. Различные аспекты, такие как безопасность, надежность и функциональность банкоматов, напрямую влияют на выбор и разработку подходящего программного обеспечения.

Банковские специалисты выполняют следующие задачи:

* управление наличными средствами в банкоматах;
* обеспечение безопасности транзакций;
* проведение технического обслуживания и мониторинга оборудования;
* настройка и обновление программного обеспечения;
* анализ и отчетность по операциям.

Промежуточные показатели, такие как статус транзакций, наличие наличных средств, время выполнения операций, могут быть получены через систему мониторинга в реальном времени. Итоговые показатели, такие как общее количество выполненных транзакций, эффективность использования банкоматов, удовлетворенность клиентов, могут быть рассчитаны на основе собранных данных.

Автоматизация процессов с помощью компьютерных систем позволяет значительно повысить эффективность работы, минимизировать вероятность ошибок, связанных с человеческим фактором, и сократить время на выполнение рутинных задач. Компьютерная обработка информации обеспечивает быстрый доступ к актуальным данным, возможность их анализа и принятия обоснованных решений на основе полученной информации. Это особенно важно в банковской сфере, где требуется оперативное реагирование на изменяющиеся условия и быстрое принятие решений.

Потенциальной аудиторией данного проекта являются банки и финансовые учреждения, их сотрудники (операционисты, IT-специалисты и менеджеры), а также клиенты, которым необходимо быстрое и удобное выполнение банковских операций через банкоматы. Проект также будет полезен для разработчиков программного обеспечения и инженеров, занимающихся обслуживанием и модернизацией банкоматов, а также для финансовых аналитиков, стремящихся оптимизировать операции и улучшить клиентский опыт.

# Постановка задачи

Исходя из анализа предметной области, можно выделить следующие задачи, подлежащие автоматизации:

* автоматизировать перевод средств с карты на карту;
* автоматизировать оплату счетов и платежей;
* автоматизировать снятие наличных средств;
* автоматизировать настройку личного счета пользователя;
* автоматизировать подач заявок на трудоустройство в банке;

В сравнении с другими приложениями для транспортной логистики, такими как «Беларусбанк», «Тинькофф», «СберБанк», «БНБ», «ПриорБанк», «АльфаБанк» и другими, ваше приложение Indusland Bank имеет следующие преимущества:

* Автоматическое управление операциями: Indusland Bank автоматически обновляет информацию о состоянии счетов и транзакциях на основе данных о банковских операциях, что помогает клиентам всегда быть в курсе своих финансов и своевременно осуществлять необходимые действия.
* Гибкость и масштабируемость: Indusland Bank спроектирован так, чтобы быть гибким и масштабируемым, позволяя легко добавлять новые функции и поддерживать большое количество пользователей. Это обеспечивает надежную работу приложения даже при увеличении числа пользователей и объемов операций.
* пользовательский интерфейс: Indusland Bank имеет интуитивно понятный и простой в использовании пользовательский интерфейс, что облегчит работу с ним для всех пользователей, независимо от их уровня технической грамотности. Это позволяет клиентам легко выполнять банковские операции и получать доступ к необходимой информации.

Таким образом, Indusland Bank предлагает уникальный набор функций, которые помогут улучшить эффективность и производительность в области банковских операций, делая его важным инструментом для любого финансового учреждения, которое хочет оставаться конкурентоспособным в современном быстро меняющемся бизнес-окружении.

# Проектирование системы

# Требования к приложению

В приложении Indusland Bank будет реализована система аутентификации для обеспечения безопасности данных пользователя. Это означает, что доступ к функционалу приложения будет возможен только после ввода уникальных учетных данных (номер карточки и пароль). Это помогает предотвратить несанкционированный доступ и защитить конфиденциальность данных пользователя.

Интерфейс приложения Indusland Bank будет разработан таким образом, чтобы обеспечить простоту и удобство использования.

Интерфейс приложения Indusland Bank будет иметь четкую и последовательную структуру. Все элементы управления будут иметь стандартный размер, форму и расположение, чтобы облегчить навигацию пользователям. Шрифт будет выбран таким образом, чтобы обеспечить хорошую читаемость. Также, цветовая схема и стиль дизайна будут выбраны так, чтобы обеспечить приятное визуальное восприятие и удобство использования.

Основными минимальными требованиями, выдвигаемыми к аппаратному обеспечению персонального компьютера, являются:

* процессор 400 МГц и выше;
* оперативная память 128 Мбайт и более;
* свободное место на диске 50 Мбайт;
* интегрированная видеокарта на 256 Мбайт и более;
* монитор;
* сенсорная панель;
* клавиатура.

# Проектирование модели

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями.

Наиболее распространенные средства моделирования являются: диаграмма вариантов использования, диаграмма классов, диаграмма последовательности, диаграмма деятельности.

Диаграмма вариантов использования – это тип диаграммы, используемой в области разработки программного обеспечения для визуализации системы с точки зрения её актёров и вариантов использования.

Основные элементы диаграммы вариантов использования:

* Актёры: Актёры представляют собой людей или системы, которые взаимодействуют с системой. Они находятся вне системы и могут быть как людьми (например, пользователи или администраторы), так и другими системами или устройствами;
* Варианты использования: Варианты использования – это функции или действия, которые система может выполнять в ответ на запросы от акторов. Они представляют собой цели, которые актёры хотят достичь с помощью системы;
* Отношения: Отношения между актёрами и вариантами использования показывают, как актёры и варианты использования взаимодействуют друг с другом. Они могут быть обозначены стрелками или линиями.

Диаграмма вариантов использования помогает разработчикам и стейкхолдерам лучше понять, как система будет работать и какие функции она будет выполнять. Она также помогает определить требования к системе и спланировать её разработку. Это важный инструмент в процессе проектирования системы.

Диаграмма вариантов использования представлена на рисунке 2.1



Рисунок 2.1 – Диаграмма вариантов

На этой диаграмме выделяются два основных пользователя. Клиент, который использует различные услуги, предоставляемые банком. Логист, который управляет данными и обработкой заявок в системе.

Диаграмма классов – это структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их атрибуты (полей), методы, интерфейсы и взаимосвязи (отношений) между ними. Она используется для визуализации структуры классов в системе и их взаимосвязей.

Вот основные элементы диаграммы классов:

* Классы: Классы представляют собой ключевые элементы в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента: имя класса, поля (атрибуты) класса и методы класса.
* Отношения: Отношения между классами на диаграмме классов могут представлять отношения, такие как ассоциация, наследование и реализация.

Диаграмма классов соответствует принципам объектно-ориентированного программирования (ООП) и является одним из базовых инструментов проектирования ООП-систем. Она помогает лучше понимать структуру системы и ее компоненты.

В данном приложении присутствуют следующие классы:

* ClientHelper;
* Helper;
* ConfirmedApplications;
* CardsTable;
* EmployyesTable;
* PDFcreator;
* OrganisationsTable.

Диаграмма классов представлена на листе 1 графической части.

Диаграмма деятельности – это графическое представление процессов и задач, выполняемых в рамках определенного проекта или деятельности. Эта диаграмма используется для анализа и оптимизации бизнес-процессов, планирования работы, принятия решений и управления проектами.

Основные элементы диаграммы деятельности:

* Начальный узел: Начальный узел активности является первым элементом диаграммы деятельности и представляет начальную точку процесса;
* Конечный узел активности: Конечный узел активности является последним элементом диаграммы деятельности и обозначает завершение процесса;
* Действие: Действия представляют собой отдельные шаги или операции, которые выполняются в процессе;
* Поток управления: Эти потоки показывают последовательность действий и передачу данных между действиями;
* Узел принятия решений: Эти узлы используются для представления ветвлений и слияний в потоке управления.

Диаграмма деятельности позволяет визуализировать последовательность задач и операций, которые нужно выполнить, чтобы достичь желаемого результата. Она является мощным инструментом для анализа, планирования и управления проектами и бизнес-процессами.

В рамках проекта рассмотрим функцию снятия наличных со счета.

Диаграмма деятельности представлена на листе 3 графической части.

Диаграмма последовательности – это UML-диаграмма, которая показывает взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций. Она отображает порядок выполнения действий и обмена информацией между объектами во времени.

Вот основные элементы диаграммы последовательности:

* Объекты: Объекты представлены горизонтальной осью диаграммы. Каждый объект имеет свою колонку.
* Сообщения: Сообщения передаются горизонтально между объектами и располагаются вертикально в хронологическом порядке. Они представлены стрелками.
* Время: Время представлено вертикальной осью диаграммы. Жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) информационной системы в рамках прецедента показаны на единой временной оси.

Диаграмма последовательности помогает анализировать и понимать взаимодействие между объектами в системе, а также порядок, в котором происходит взаимодействие. Она является важным инструментом для анализа и проектирования систем.

Диаграмма последовательности представлена на листе 4 графической части.

# Организация данных

Наиболее распространенным средством моделирования таких типов документации являются схемы-сети (ERD), которые применяются для графического представления многих элементов разработанных программ систем и представляют некоторые габаритные размеры обозначений для определения данных и отношений между ними. С помощью такой диаграммы можно описать отдельные компоненты концептуальной модели или выразить свою точку зрения взаимосвязей между ними, иногда какое значение или реализацию системы. Основными потенциалом данной нотации является понятная сущность в каждом случае каждый рассматриваемый объект может явиться экземпляром одного и только одного сущности, должен иметь уникальное имя или идентификатор, а также отличается от других экземпляров данной сущности. Следовательно: как фотонные линии короткого ассортимента между отделами сущности.

Графическая модель данных строится таким образом, чтобы связь между отделениями сущности органа не только естественный характер соответствующего отношения, но и дополнительные аспекты обязательности связей.

Информационная модель представлена на диаграмме «Сущность-связь» на рисунке 2.2

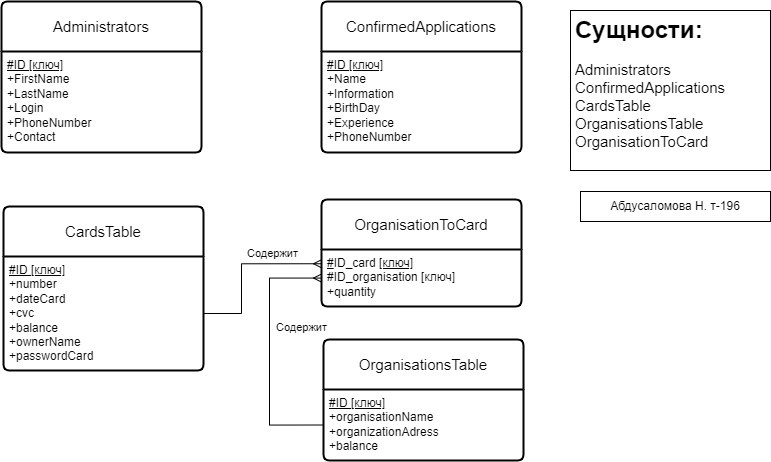


Рисунок 2.2 – Диаграмма «Сущность-связь»

Исследовав предметную область, можно выделить следующие сущности: «Administrators», «CardsTable», «ConfirmedApplications», «OrganisationToCard», «OrganisationTable».

Для сущности «Administrators» атрибутам будут являться:

* номер;
* имя;
* фамилия;
* логин;
* пароль;
* контактные данные.

Для сущности «CardsTable» атрибутами будут являться:

* номер;
* номер карточки;
* номер платежной системы CVC;
* баланс;
* имя держателя;
* пароль.

Для сущности «ConfirmedApplications» атрибутами будут являться:

* номер;
* имя клиента;
* подробная информация;
* день рождения;
* стаж работы;
* номер телефона.

Для сущности «OrganisationToCard» атрибутами будут являться:

* идентификационный номер карточки;
* идентификационный номер организации;
* количество.

Для сущности «OrganisationTable» атрибутами будут являться:

* номер;
* адрес организации;
* название организации;
* баланс.

Типы связей между сущностями:

* организация – карта-организация: один ко многим;
* карта – карта-организация: один ко многим.

# Концептуальный прототип

Концептуальный прототип состоит из описания внешнего вида пользовательского интерфейса, а именно, элементов управления.

Разрабатываемое приложение будет содержать три основные формы: «Клиентская часть», «Администраторская часть» и «Вход».

На форме «Вход» есть два текстовых поля: номер карточки и пароль. При некорректном вводе, либо вводе несоответствующих значений пользователю показываются окна, содержащие суть ошибки и путь ее обхода. При нажатии кнопки «Войти», при условии корректности введенных данных, открывается основное клиентское окно. При нажатии кнопки «Войти как администратор», откроется основное окно администратора.

Интерфейс форм представлен в приложении Б.

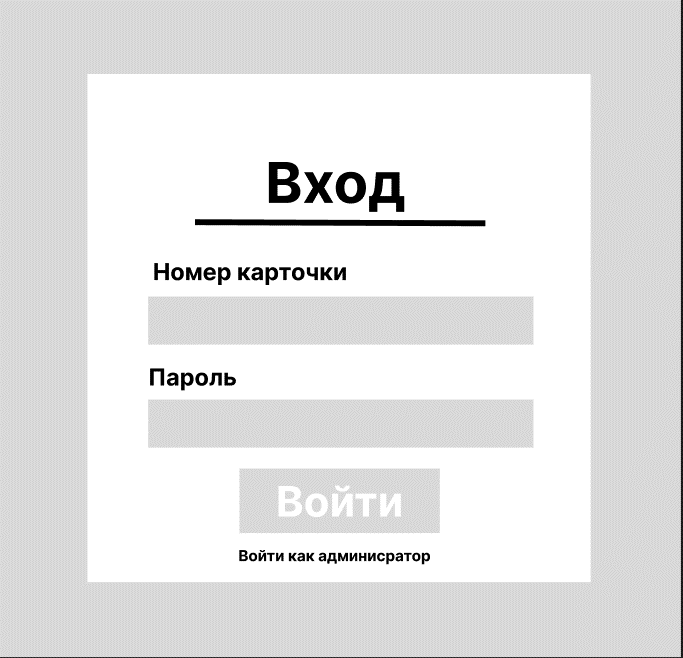


Рисунок 2.3 – Форма входа

На основной форме находятся следующие элементы:

* **Кнопка «Выйти»: при нажатии закрывают программу;**
* **Элемент с балансом и номер карточки, на которой находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Кнопка «Стать нашим сотрудником»: при нажатии открывается форма, в которой заполняется заявка;**
* **Кнопка «Снять наличные со счета»: при нажатии открывается форма, с помощью которой можно снять деньги со счета в виде наличных средств;**
* **Кнопка «Платёжные операции»: при нажатии открывается форм, на которой можно совершить различные платёжные операции;**
* **Кнопка «Перевод средств»: при нажатии открывается форма перевода средств на иной счет;**
* **Кнопка «Настройки карты»: при нажатии открывается форма, где можно изменить некоторые параметры банковской карты;**
* **Кнопка «Информация о карточке»: показывает окно с информацией о карте.**
* **Кнопка «Наши партнеры»: при нажатии открывается окно, на которой показаны партнеры банка.**

Интерфейс представлен на картинке 2.4



Рисунок 2.4 – Главное окно клиентской части

**Окно «Стать нашим сотрудником» состоит из следующих элементов**:

* **Кнопка «Отмена»: при нажатии возвращает пользователя на главное окно;**
* **Элемент с балансом и номер карточки, на которой находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Поле «ФИО»: в это поле пользователь вводит свои фамилия, имя и отчество;**
* **Поле «О себе»: в это поле пользователь вводит свою краткую биографию, опыт работы и так далее;**
* **Поле «Стаж работы»: в это поле пользователь вводит свой рабочий стаж;**
* **Поле «Дата рождения»: в это поле пользователь вводит свою дату рождения;**
* **Поле «Номер телефона»: в это поле пользователь вводит свой номер телефона, по которому с ним можно будет связаться позже;**
* **Кнопка «Подать заявку»: при нажатии заявка сохраняется;**

**Интерфейс представлен на рисунке 2.5**

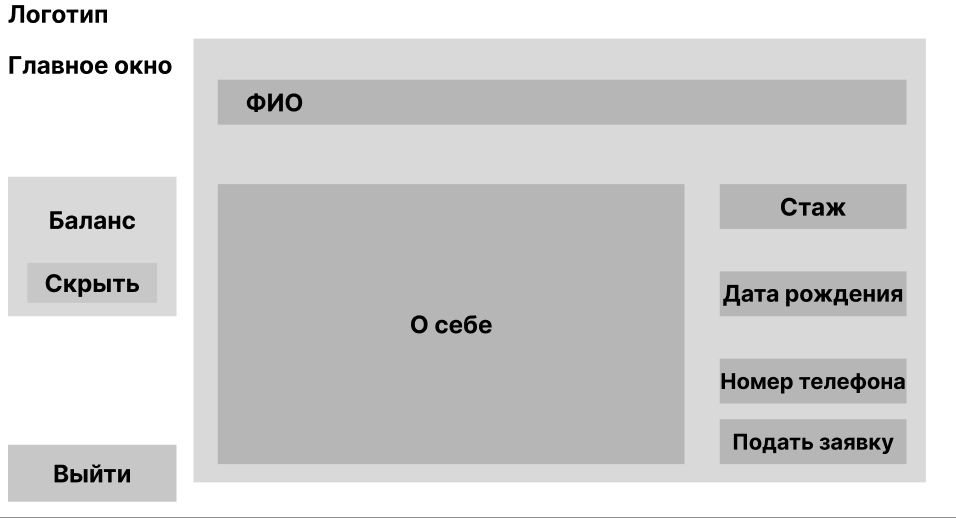


Рисунок 2.5 – Форма добавления посылки

На форме «Снятие наличных» находятся следующие элементы:

* **Кнопка «Назад»: при нажатии возвращает пользователя на главное окно;**
* **Элемент с балансом и номер карточки, на которой находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Выпадающий список, в котором находятся возможные валюты для перевода – BYN, EUR, USD.**
* **Кнопки с уже определенными суммами перевода – 5, 10, 20, 50 и 100 у.е.;**
* **Поле «Иная сумма» в которую пользователь может ввести произвольную сумму снятия;**
* **Кнопка «Подтвердить»: при нажатии кнопки, при условии корректности данных и положительного баланса, перевод будет осуществлен.**

Интерфейс представлен на рисунке 2.6



Рисунок 2.6 – Окно вывода наличных средств

Интерфейс формы «Оплата платежей» состоит из нескольких этапов. Первый этап – выбор типа платежа, на которой находятся следующие элементы:

* **Кнопка «Назад»: при нажатии осуществляется возврат на главную страницу;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Кнопка «Мобильная связь»: при нажатии открывается возможность оплаты по номеру операторов life, a1, МТС;**
* **Кнопка «ЖКХ / ЖКУ»: при нажатии открывается возможность оплаты жилищно-коммунальных услуг;**
* **Кнопка «Интернет, телевидение»: при нажатии открывается возможность оплаты домашнего и не только интернета;**
* **Кнопка «Оплата налогов»: при нажатии открывается возможность оплаты налога на прибыль, акцизов, налога на недвижимость и других налогов;**
* **Кнопка «Оплата штрафов»: при нажатии открывается возможность оплаты штрафов за нарушение ПДД, ДЦП и другие штрафы;**
* **Кнопка «Образование и развитие»: при нажатии открывается возможность оплаты автошкол, ВУЗов, школьного питания, ПТУ и других образовательных услуг.**

**Интерфейс представлен на рисунке 2.7.1**



Рисунок 2.7.1 – Форма выбора типа платежа

Второй этап – ввод адреса и суммы получателя. Данная форма состоит из следующих элементов:

* **Кнопка «Назад»: при нажатии осуществляется возврат к выбору услуги;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Поле «Введите адрес получателя»: пользователь вводит адрес получателя;**
* **Поле «Введите сумму»: пользователь вводит сумму**
* **Выпадающий список «Введите валюту»: пользователь выбирает валюту – BYN, EUR, USD;**
* **Кнопка «Подтвердить»: при нажатии кнопки, при условии корректности данных и положительного баланса, оплата будет осуществлена.**

**Интерфейс представлен на рисунке 2.7.2**

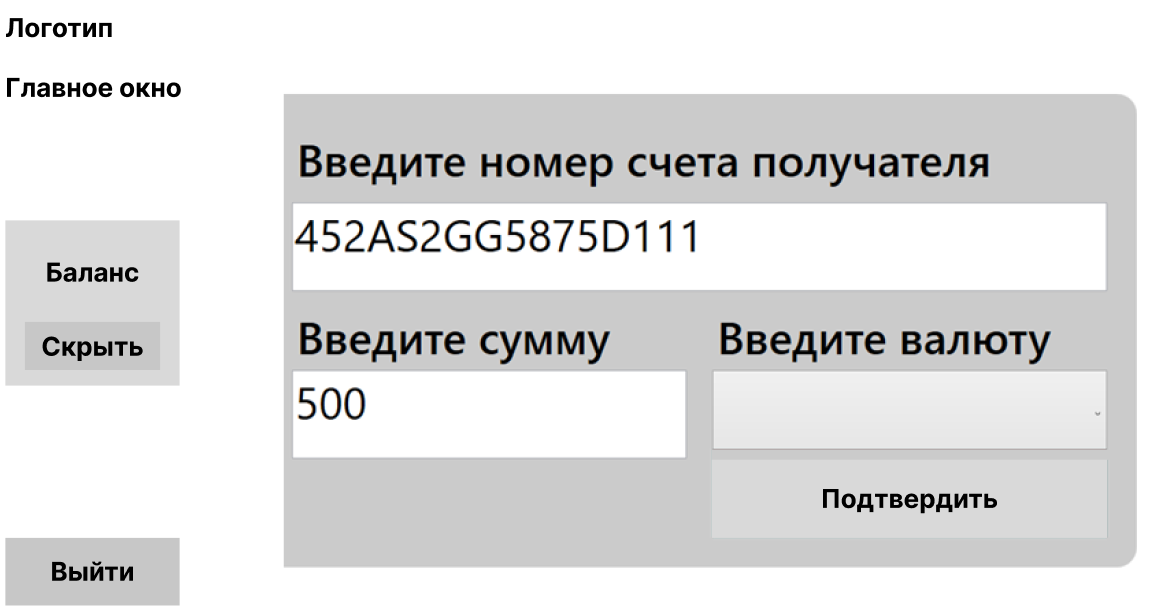


Рисунок 2.7.2 – Форма ввода адреса и суммы

Интерфейс формы «Перевод с карты на карту» состоит из нескольких этапов. Первый вариант – ввод номера карты получателя. Эта форма состоит из:

* **Кнопка «Назад»: при нажатии осуществляется возврат в главное окно;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Поле «Введите номер карты получателя»: пользователь вводит номер карты получателя;**
* **Цифровая клавиатура для ввода номера карты;**
* **Кнопка подтверждения ввода номера карты (галочка).**

**Интерфейс представлен на рисунке 2.8.1**



Рисунок 2.8.1 – Форма ввода номера карты пользователя

Второй этап – ввод суммы отправления. Эта форма состоит из:

* **Кнопка «Назад»: при нажатии осуществляется возврат на прошлый этап;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Поле «Введите сумму отправления»: пользователь вводит сумму перевода;**
* **Цифровая клавиатура для ввода номера карты;**
* **Кнопка подтверждения ввода суммы перевода и валюты («BYN» и «USD»).**

**Интерфейс представлен на рисунке 2.8.2**

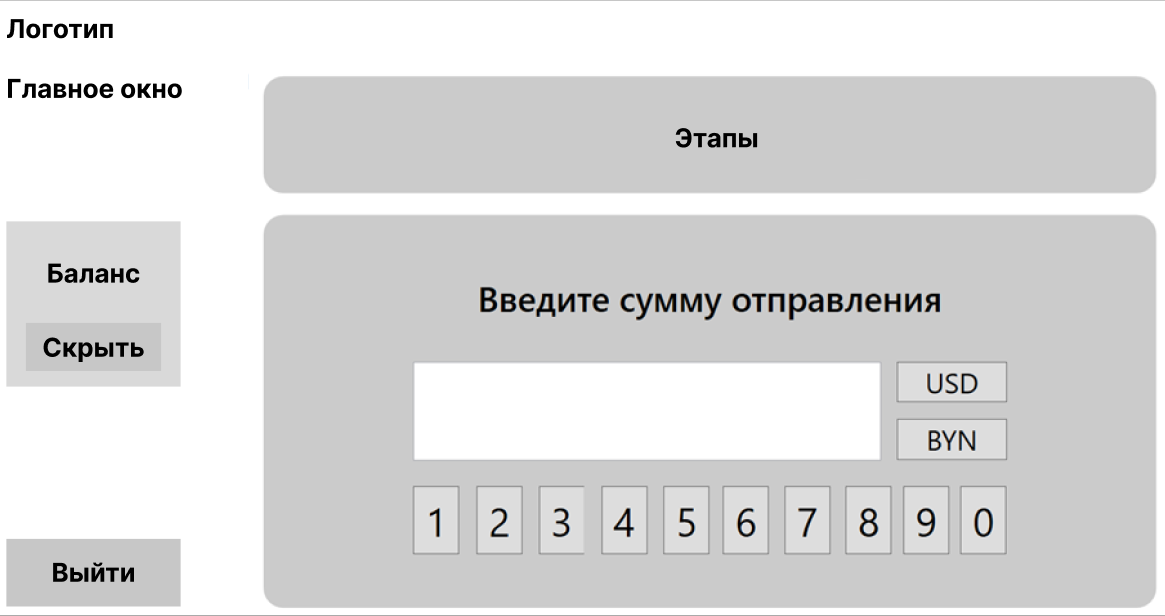


Рисунок 2.8.2 – Форма ввода суммы и валюты перевода

Третий этап – подтверждение перевода. Эта форма состоит из:

* **Кнопка «Назад»: при нажатии осуществляется возврат на второй этап;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Информация перевода (номер карты получателя и отправителя, сумма перевода)**
* **Кнопка «Отмена», при нажатии которой операция будет отменена;**
* **Кнопка «Подтвердить», при нажатии которой операция будет произведена.**

**Интерфейс представлен на рисунке 2.8.3**

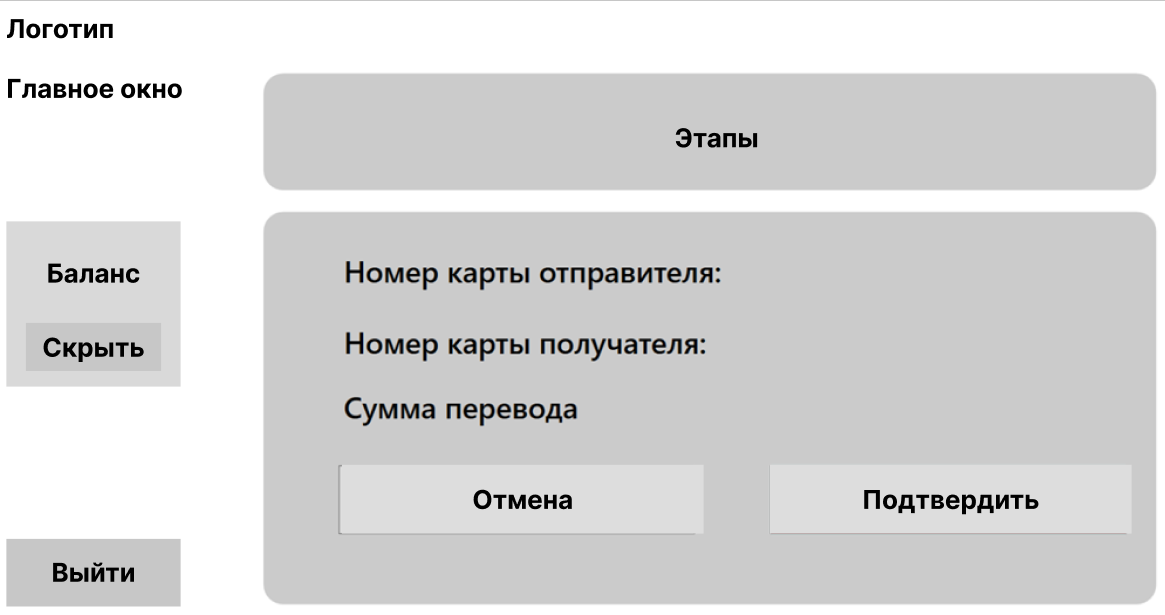


Рисунок 2.8.3 – Форма подтверждения перевода

Интерфейс «Настройки карты» состоит из:

* **Кнопка «Назад»: при нажатии осуществляется возврат на второй этап;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Кнопка «Закрыть свой счет»: при нажатии которой нужно будет подтвердить действия. при положительном ответе карта перестает быть дееспособной;**
* **Кнопка «Продлить срок действия карты»: после нажатия, нужно будет ввести новую дату и подтвердить действие (за данную услугу взимается оплата в размере 15 белорусских рублей);**
* **Кнопка «Изменение пароля»: при нажатии которой, нужно будет ввести новый пароль, при корректности данных и достатке средств на счете, пароль изменится (стоимость услуги – 15 белорусских рублей);**
* **Кнопка «Получение номера CVC\CVV»: по нажатию которой, необходимо будет ввести пароль от карты чтобы получить уникальный номер платежной системы (услуга бесплатна).**

**Интерфейс представлен на рисунке 2.9**

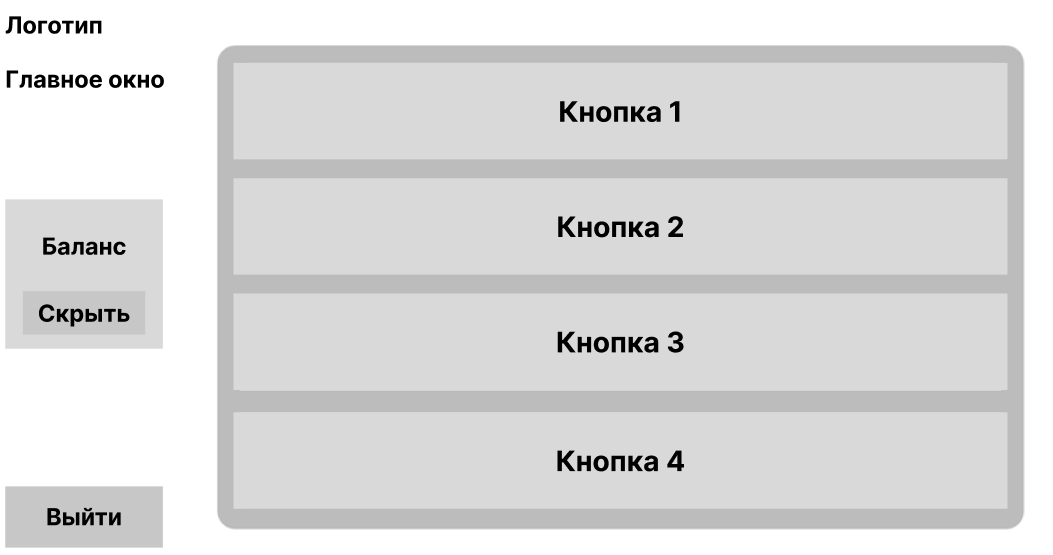


Рисунок 2.9 – Форма настройки карты

Интерфейс формы «Наши партнеры» состоит из:

* **Кнопка «Назад»: при нажатии осуществляется возврат на второй этап;**
* **Элемент с балансом и номер карточки, на котором находится кнопка «Скрыть остаток», при нажатии которой скрывается остаток счета на балансе, при повторном нажатии, остаток показывается;**
* **Логотипы наших партнеров.**

**Интерфейс представлен на рисунке 2.10**

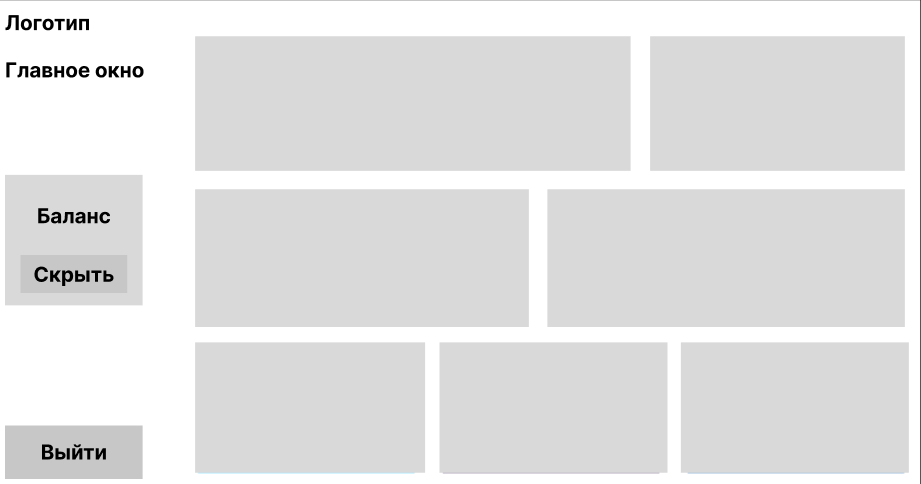


Рисунок 2.10 – Форма партнеров

Интерфейс основной формы администратора состоит из:

* **Кнопка «Клиенты», по нажатию которой будет показана страница с информацией о всех зарегистрированных клиентах банка;**
* **Кнопка «Организации», по нажатию которой будет показана страница с информацией о всех зарегистрированных организациях банка;**
* **Кнопка «Сотрудники», по нажатию которой будет показана страница с информацией о всех сотрудниках банка;**
* **Кнопка «Приложения», по нажатию которой будет показана страница с информацией о всех заявках от клиентов;**
* **Кнопка «Выйти», по нажатию которой, окно будет закрыто.**

**Интерфейс представлен на рисунке 2.11**



Рисунок 2.11 – Основная форма администратора

Интерфейс формы списка клиентов состоит из:

* **Таблица данных, в которой представлена информация о клиентах;**
* **Кнопка «Добавить», при нажатии которой можно будет добавить запись в таблицу;**
* **Кнопка «Удалить», при нажатии которой можно будет удалить запись из таблицы;**
* **Кнопка «Обновить», при нажатии которой страница будет обновлена;**
* **Кнопка «Поиск», при нажатии которой будет возможен поиск по ключевому полю.**

**Интерфейс представлен на рисунке 2.12**

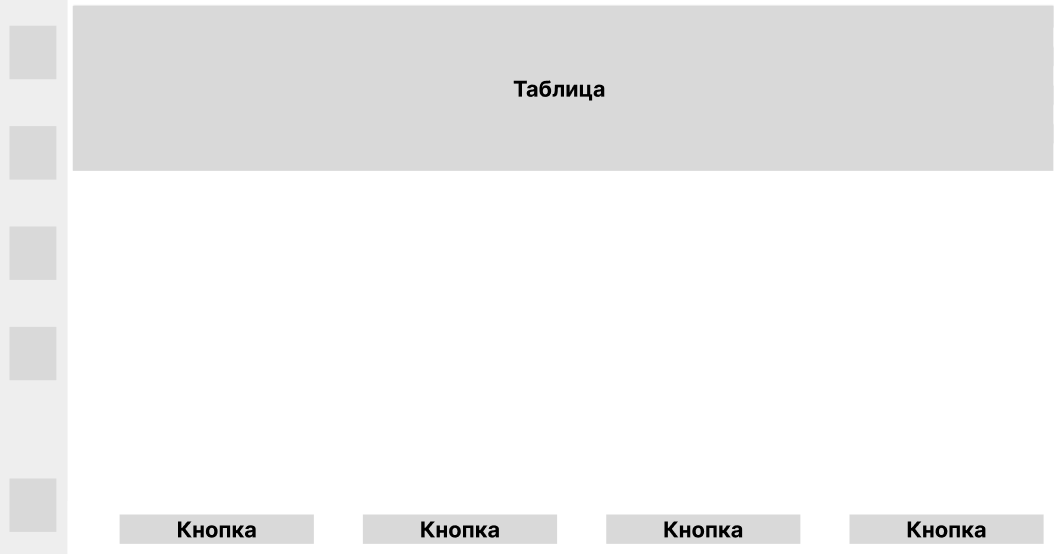


Рисунок 2.12 – Форма списка клиентов

Интерфейс формы списка организаций состоит из:

* **Таблица данных, в которой представлена информация об организациях;**
* **Кнопка «Добавить», при нажатии которой можно будет добавить запись в таблицу;**
* **Кнопка «Удалить», при нажатии которой можно будет удалить запись из таблицы;**
* **Кнопка «Обновить», при нажатии которой страница будет обновлена;**
* **Кнопка «Поиск», при нажатии которой будет возможен поиск по ключевому полю.**

**Интерфейс представлен на рисунке 2.13**

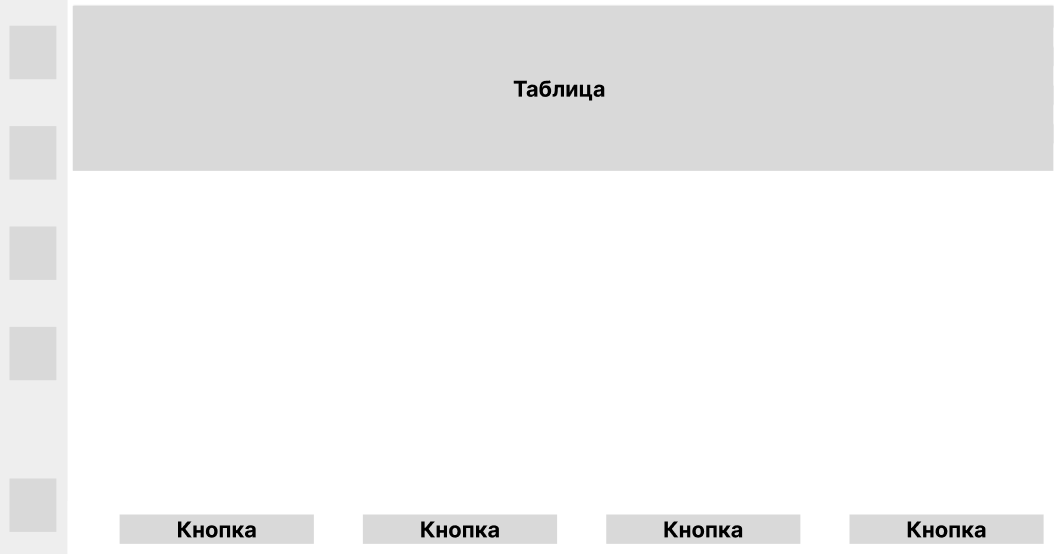


Рисунок 2.13 – Форма списка организаций

Интерфейс формы списка сотрудников состоит из:

* **Таблица данных, в которой представлена информация о сотрудниках;**
* **Кнопка «Добавить», при нажатии которой можно будет добавить запись в таблицу;**
* **Кнопка «Удалить», при нажатии которой можно будет удалить запись из таблицы;**
* **Кнопка «Обновить», при нажатии которой страница будет обновлена;**
* **Кнопка «Поиск», при нажатии которой будет возможен поиск по ключевому полю.**

Интерфейс представлен на рисунке 2.14

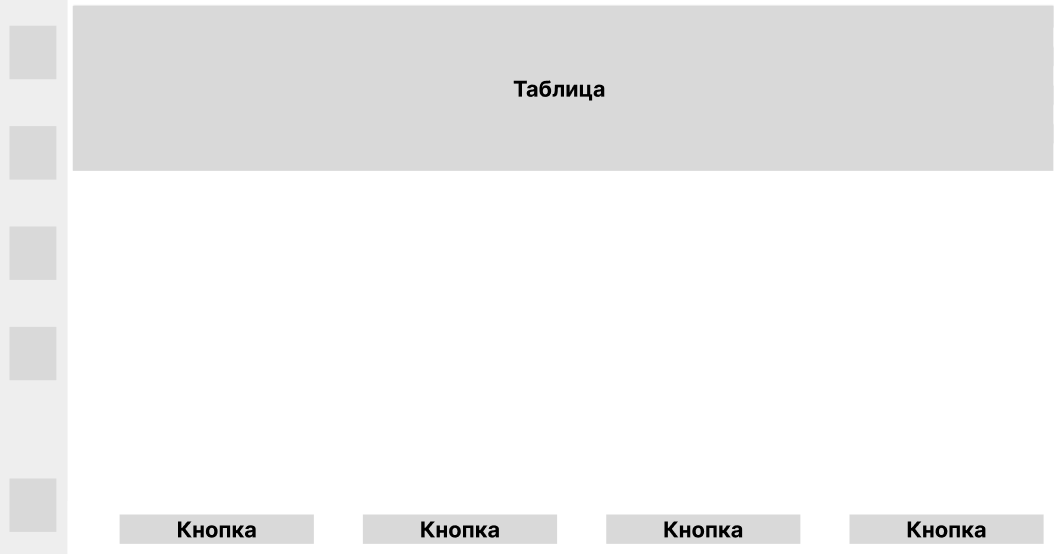


Рисунок 2.14 – Форма списка сотрудников

Интерфейс формы приложений состоит из:

* **Список «Новые заявки», в котором показываются все заявки, оставленные пользователями;**
* **Поле «ФИО» для отображения полного имени, фамилии и отчества;**
* **Поле «Стаж работы» для отображения опыта работы;**
* **Поле «Дата рождения» для отображения даты рождения;**
* **Поле «Номер телефона» для отображения номера телефона;**
* **Поле «Подробная информация», в котором находится более подробная информация о клиенте;**
* **Кнопка «Принять» для принятия заявки;**
* **Кнопка «Отклонить» для отказа на заявку;**

Интерфейс представлен на рисунке 2.15

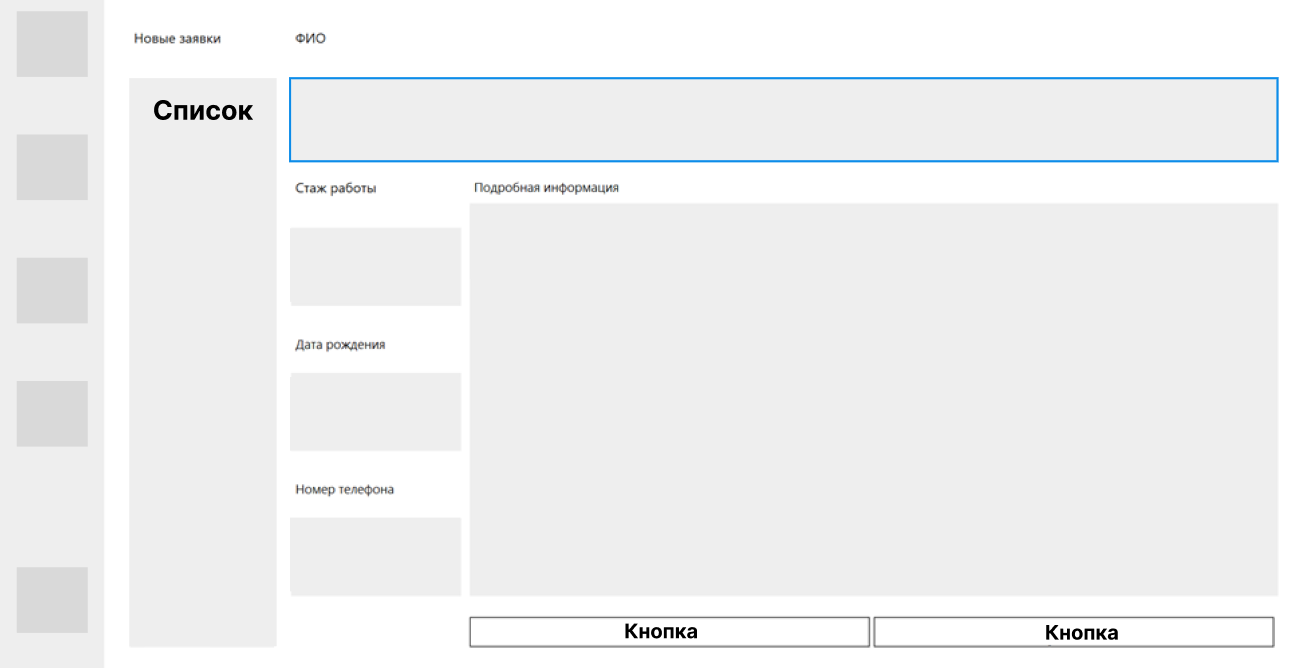


Рисунок 2.15 – Форма приложений

# Описание реализации программного средства

# Инструменты разработки и применения технологии

Инструментами разработки для будущего программного приложения будут являться:

* операционная система MS Widows 10 и выше;
* программная среда разработки Microsoft Visual Studio 2022;
* язык программирования C#;
* система управления реляционными базами данных (СУБД) Microsoft SQL Server 2024;
* офисный пакет приложений Microsoft Office.

MS Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства MS Windows. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами.

Microsoft Visual Studio 2022 – это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию.

MS Visual Studio 2022 включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня.

Microsoft SQL Server – система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

# Порядок авторизации пользователей

В приложении можно выделить следующие роли и функции:

Клиент банка:

* Аутентификация: Клиент регистрируется в системе, предоставляя необходимые учетные данные (например, номер карточки и пароль);
* Авторизация: После успешной аутентификации клиент банка получает доступ к основным функциям;
* Функции: Перевод средств с карты на карту, снятие наличных средств, оплата платежей, подача заявок, изменения персональных данных, получение информации о карточке и просмотр организаций-партнеров.

Администратор:

* Аутентификация: Администратор регистрируется в системе, предоставляя необходимые учетные данные;
* Авторизация: После успешной аутентификации администратор получает доступ к функциям;
* Функции: Добавление, удаление и поиск среди клиентов, организаций и сотрудников банка, также рассмотрение, поданных клиентами, заявок.

# Организация данных

Таблица «ConfirmedApplication» хранит информацию о пользователях, подавших заявки на работу 3.1

Таблица 3.1 – Структура таблицы «ConfirmedApplication»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| ID | INTEGER | 8 | Идентификационный номер |
| Name | nvarchar(MAX) | n\*2 | Название имя клиента |
| Information | nvarchar(MAX) | n\*2 | Подробная информация о клиенте |
| BirthDay | DateTime | 8 | День рождения клиента |
| Experience | nvarchar(MAX) | n\*2 | Стаж работы |
| PhoneNumber | nvarchar(MAX) | n\*2 | Номер телефона |

Таблица «Administrators» хранит информацию о всех администраторах. Структура приведена в таблице 3.2

Таблица 3.2 – Структура таблицы «Administrators»

|  |  |  |  |
| --- | --- | --- | --- |
| Название строки | Тип данных | Размер в байтах | Описание |
| Id | INTEGER | 8 | Идентификационный номер |
| FirstName | varchar(50) | 100 | Имя администратора |
| LastName | varchar(50) | 100 | Фамилия администратора |
| Login | varchar(50) | 100 | Логин |
| Password | varchar(255) | 510 | Пароль |

Продолжение таблицы 3.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Contact | varchar(100) | 200 | Контактные данные |

Таблица «CardsTable» хранит информацию о клиентах, структура приведена в таблице 3.3

Таблица 3.3 – Структура таблицы «CardsTable»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | INTEGER | 8 | Идентификационный номер |
| number | nchar(16) | 32 | Номер карточки |
| dateCard | nchar(4) | 8 | Дата действия |
| cvc | nchar(3) | 6 | Уникальный номер |
| passwordCard | nchar(4) | 8 | Пароль |
| ownerName | nchar(40) | 80 | Имя держателя карты |
| balance | money | 14 | Баланс карты |

Таблица «organisationsTable» хранит информацию о организациях, структура приведена в таблице 3.4

Таблица 3.4 – Структура таблицы «organisationsTable»

|  |  |  |  |
| --- | --- | --- | --- |
| Название строки | Тип данных | Размер в байтах | Описание |
| 1 | 2 | 3 | 4 |
| Имя поля | Тип данных | Размер, байт | Описание |
| Id | INTEGER | 8 | Идентификационный номер |
| organisationName | nchar(50) | 100 | Название организации |
| organizationAdress | nchar(50) | 100 | Адрес организации |

Таблица «OrganisationToCard» хранит информацию о связях между картами и организациями, структура приведена в таблице 3.5

Таблица 3.5 – Структура таблицы «OrganisationToCard»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер, байт | Описание |
| ID\_card | INTEGER | 8 | Вторичный ключ, указывающий на карту |
| ID\_organisation | INTEGER | 8 | Вторичный ключ, указывающий на организацию |
| quantity | INTEGER | 8 | Количество |

Все таблицы представлены на рисунке 3

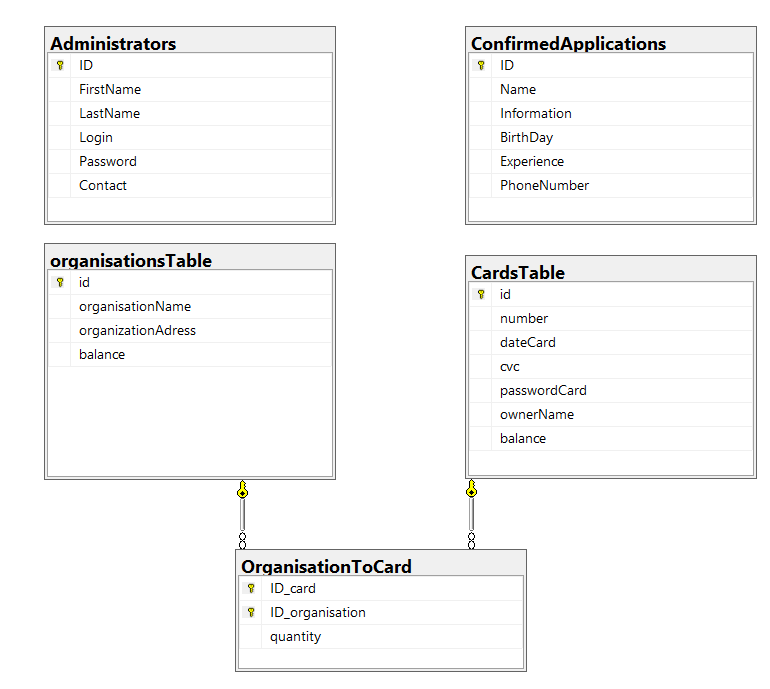


Рисунок 3 – Схема данных

# Функции: обоснование выбранной организации

В данном проекте реализованы функции добавления, поиска и удаления информации о клиентах, организациях и сотрудниках; рассмотрение заявок, оставленных пользователями.

Функция добавления клиента в систему находиться на форме «Список клиентов». После заполнения соответствующих полей и нажатия на кнопку «Добавить» на данной форме происходит обработка события addButton\_Click. Все данные, вводимые в поля формы, будут сохранены в таблице «CardsTable».

Функция поиска клиента находится на форме «Список клиентов». После заполнения соответствующего поля с ключевым значением и нажатия на кнопку «Найти» на данной форме происходит обработка событий searchButton\_Click, после чего будут показаны результаты поиска.

Функция удаления информации о клиенте находится на форме «Список клиентов». После выбора клиента в списке и нажатия на кнопку «Удалить» на данной форме происходит обработка событий deleteButton\_Click. Выбранные данные будут удалены из таблицы «CardsTable».

Функция добавления организации в систему находиться на форме «Список организаций». После заполнения соответствующих полей и нажатия на кнопку «Добавить» на данной форме происходит обработка события addButton\_Click. Все данные, вводимые в поля формы, будут сохранены в таблице «organisationsTable».

Функция поиска организации находится на форме «Список организаций». После заполнения соответствующего поля с ключевым значением и нажатия на кнопку «Найти» на данной форме происходит обработка событий searchButton\_Click, после чего будут показаны результаты поиска.

Функция удаления организации находится на форме «Список организаций». После выбора организации в списке и нажатия на кнопку «Удалить» на данной форме происходит обработка событий deleteButton\_Click. Выбранные данные будут удалены из таблицы «organisationsTable».

Функция добавления сотрудника в систему находиться на форме «Список сотрудников». После заполнения соответствующих полей и нажатия на кнопку «Добавить» на данной форме происходит обработка события addButton\_Click. Все данные, вводимые в поля формы, будут сохранены в таблице «Administrators».

Функция поиска сотрудника находится на форме «Список сотрудников». После заполнения соответствующего поля с ключевым значением и нажатия на кнопку «Найти» на данной форме происходит обработка событий searchButton\_Click, после чего будут показаны результаты поиска.

Функция удаления сотрудника находится на форме «Список сотрудников». После выбора сотрудника в списке и нажатия на кнопку «Удалить» на данной форме происходит обработка событий deleteButton\_Click. Выбранные данные будут удалены из таблицы «Administrators».

Функция рассмотрения поданных заявок находится на форме «Приложения», на которой, после выбора заявки в списке, будет показаны следующие данные: ФИО, стаж работы, дата рождения, номер телефона и подробная информация.

Функция отклонения заявления находится на форме «Приложения». После выбора определенной заявки и нажатия на кнопку «Отклонить» на данной форме, происходит обработка событий deny\_Button\_Click. Выбранные данные будут удалены из системы.

Функция принятия заявки находится на форме «Приложения». После выбора определенной заявки и нажатия на кнопку «Принять» на данной форме, происходит обработка событий apply\_Button\_Click. Выбранные данные будут перенесены в таблицу «ConfirmedApplication».

Полный код программы представлен в приложении А.

# Входные и выходные данные

Входными данными являются данные, заносимые программой в таблицы: «CardsTable», «Administrators», «organisationsTable», «ConfirmedApplication». Также входными данными являются данные, вводимые в поля для поиска.

Для добавления данных в таблицу «CardsTable» необходимо ввести следующие данные:

* номер карточки;
* срок действия;
* имя держателя.

Для добавления данных в таблицу «Administrators» необходимо ввести следующие данные:

* имя;
* фамилия;
* логин;
* пароль;
* контактная информация.

Для добавления данных в таблицу «organisationsTable» необходимо ввести следующие данные:

* название организации;
* адрес организации.

Для добавления данных в таблицу «ConfirmedApplication» необходимо ввести следующие данные:

* ФИО;
* подробная информация;
* день рождения;
* стаж работы;
* номер телефона.

Выходные данными являются данные, которые экспортируются таблицу «ConfirmedApplication», а именно принятые заявки.

Выходные документы представлен в приложении Б.

# Функциональное тестирование

Функциональное тестирование – это тестирование функций приложения на соответствия требованиям. Оценка производится в соответствии с ожидаемым и фактическим результатом.

Тестирование программы будет производится последовательно, переходя из одной программы в другую. Во время теста будут проверятся все действия с программой, навигация пунктов меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

В таблицах 3.3-3.14 представлены тест-кейсы, подготовленные для проведения функционального тестирования.

Первый тест будет проведён в окне авторизации программы.

Тест-кейс для тестирования функции авторизации представлен в таблице 3.3

Таблица 3.3 – Тест-кейс функции авторизации

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 1 | Вход в аккаунт | 1. Ввода номера карточки «5428965845216954» 2. Ввод пароля «5240» 3. Нажать «Войти» | Ожидаемый результат: вход в аккаунт и переход на главную страницу |
| Фактический результат: успешный вход в аккаунт и переход на главную страницу |

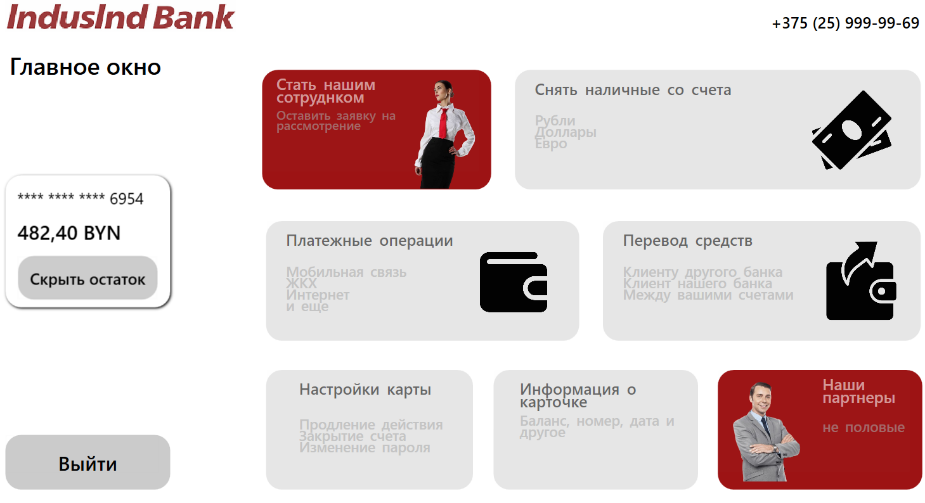


Рисунок 3.3 – Результат тест-кейс

Второй тест будет проведён на форме «Стать нашим сотрудником».

Тест-кейс для тестирования функции подачи заявки будет представлен в таблицы 3.4

Таблица 3.4 – Тест-кейс функции подачи заявки

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 2 | Подача заявки | 1. Ввода ФИО «Иванов Иван Иванович» 2. Ввести информацию о себе «целеустремленный, быстро адаптируюсь к работе» 3. Ввод стажа «10» 4. Ввод даты рождения «10.12.2000» 5. Ввод номера телефона «375258888888» 6. Нажать «подать заявку» | Ожидаемый результат: появиться сообщение об успешной отправке заявки. |
| Фактический результат: сообщение «ваша заявка принята» |

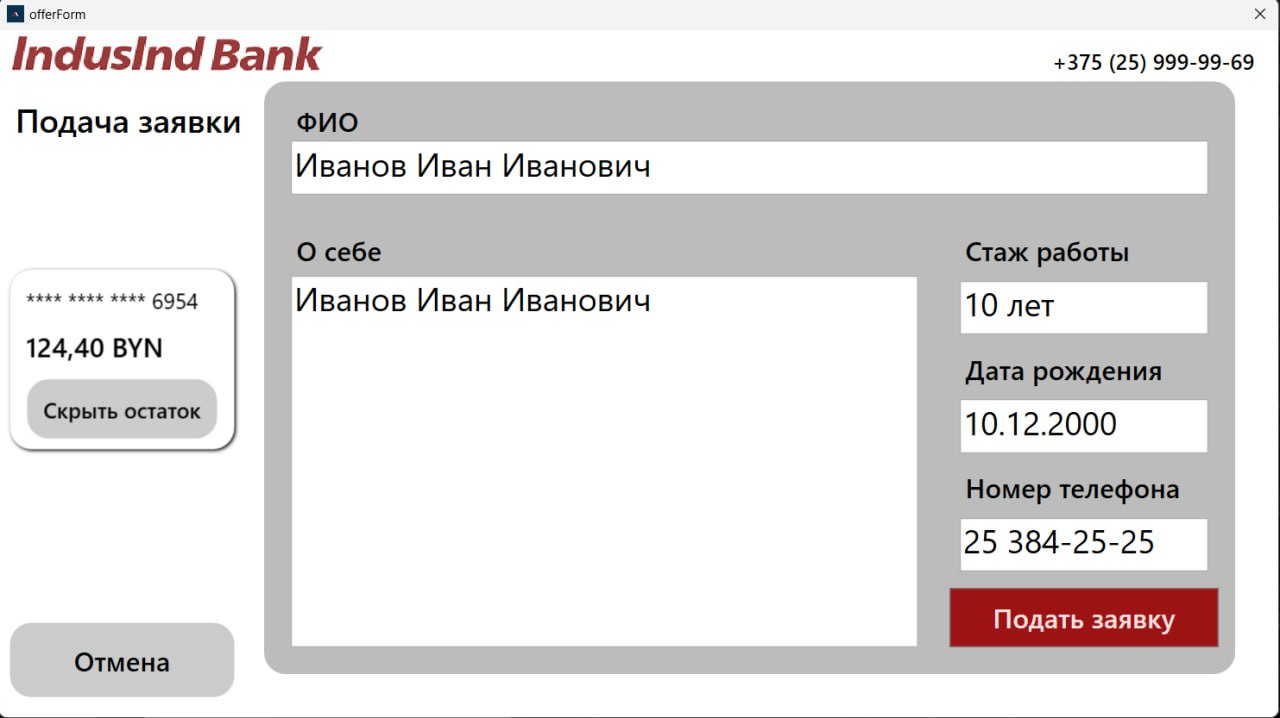


Рисунок 3.4 – Результат тест-кейс

Третий тест будет проведён на форме «Снятие наличных».

Тест-кейс для тестирования функции снятия наличных будет представлен в таблицы 3.5

Таблица 3.5 – Тест-кейс функции снятия наличных

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 3 | Снятие наличных | 1. Выбрать или ввести сумму вывода 2. Выбрать валюту 3. Нажать кнопку «Подтвердить» | Ожидаемый результат: появиться сообщение об успешном снятие наличных |
| Фактический результат: сообщение «деньги снята», снятие денег со счета |

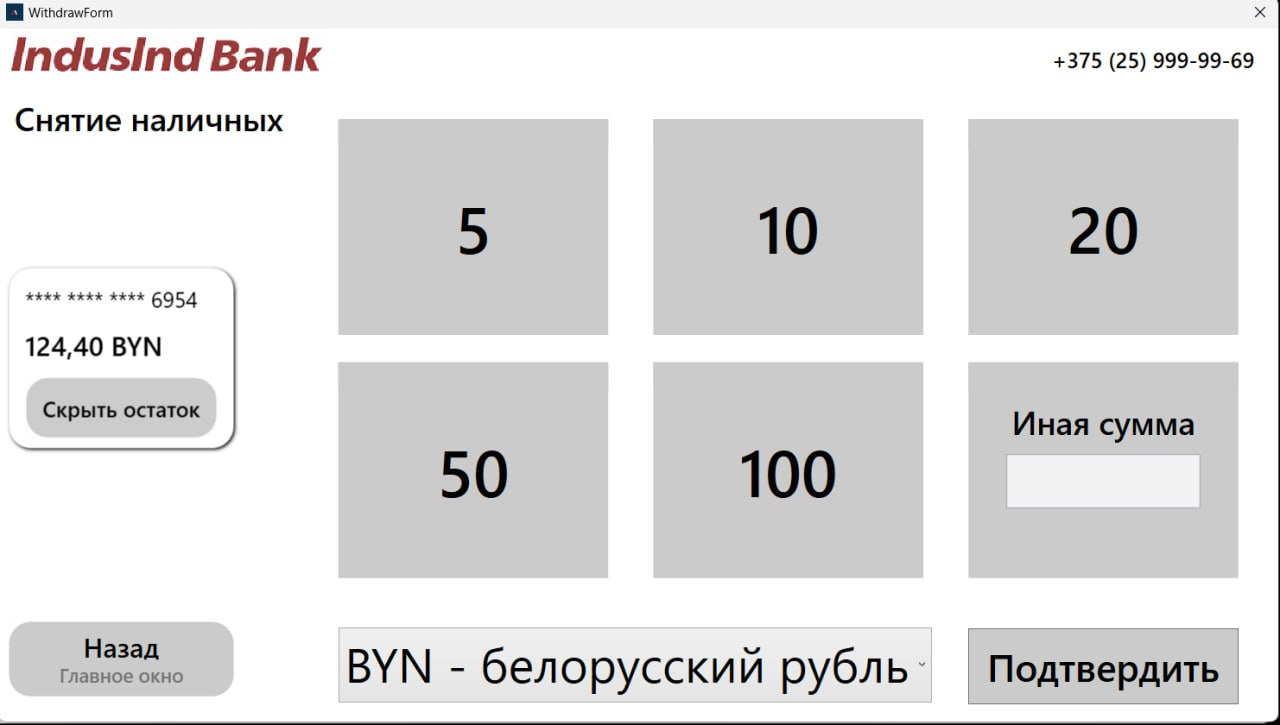


Рисунок 3.5 – Результат тест-кейс

Четвертый тест будет проведён на форме «Основная форма» для клиентов.

Тест-кейс для тестирования функции скрытия остатка будет представлен в таблицы 3.6

Таблица 3.6 – Тест-кейс функции скрытия остатка

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 4 | Скрыть остаток | 1. Нажать «скрыть остаток» | Ожидаемый результат: остаток баланса будет скрыт |
| Фактический результат: остаток баланса будет скрыт |



Рисунок 3.6 – Результат тест-кейс

Пятый тест будет проведён на форме «Оплата платежей».

Тест-кейс для тестирования функции оплата платежей будет представлен в таблицы 3.7

Таблица 3.7 – Тест-кейс функции оплата платежей

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 5 | Оплата платежей | 1. Введите номер счета получателя ‘452AS2GG5875D111’. 2. Укажите сумму перевода ‘500’. 3. Выберите валюту перевода ‘EUR - евро’. 4. Проверьте правильность введенных данных. 5. Нажмите кнопку ‘Подтвердить’ для инициации транзакции. | Ожидаемый результат: сумма списывается с вашего счета и зачисляется на счет получателя. |
| Фактический результат: сообщение «оплата прошла успешно» |

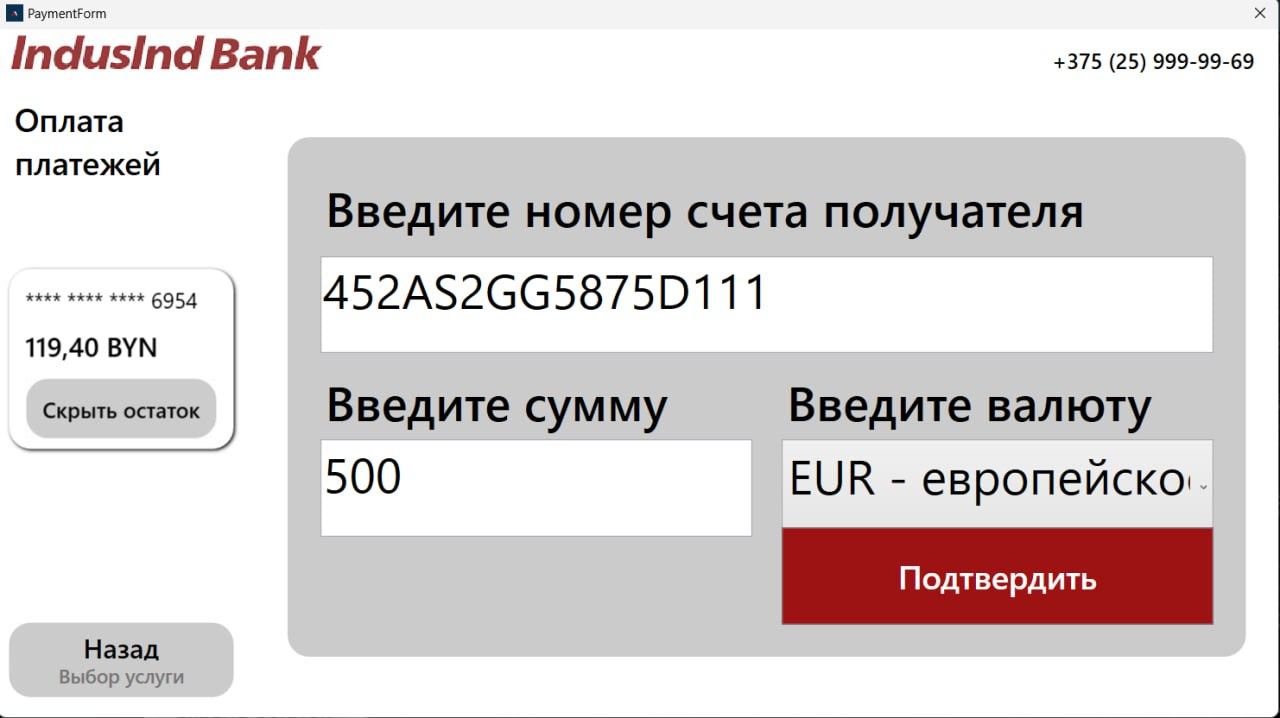


Рисунок 3.7 – Результат тест-кейс

Шестой тест будет проведён на форме «Перевод с карты на карту».

Тест-кейс для тестирования функции перевод с карты на карту будет представлен в таблицы 3.8

Таблица 3.8 – Тест-кейс функции перевод с карты на карту

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 6 | Перевод на карту средств | 1. Введите номер карты получателя. 2. Нажмите кнопку ‘Подтвердить’ для инициации транзакции. | Ожидаемый результат: переход на окно «ввод суммы» |
| Фактический результат: успешный переход на окно «ввод суммы» |

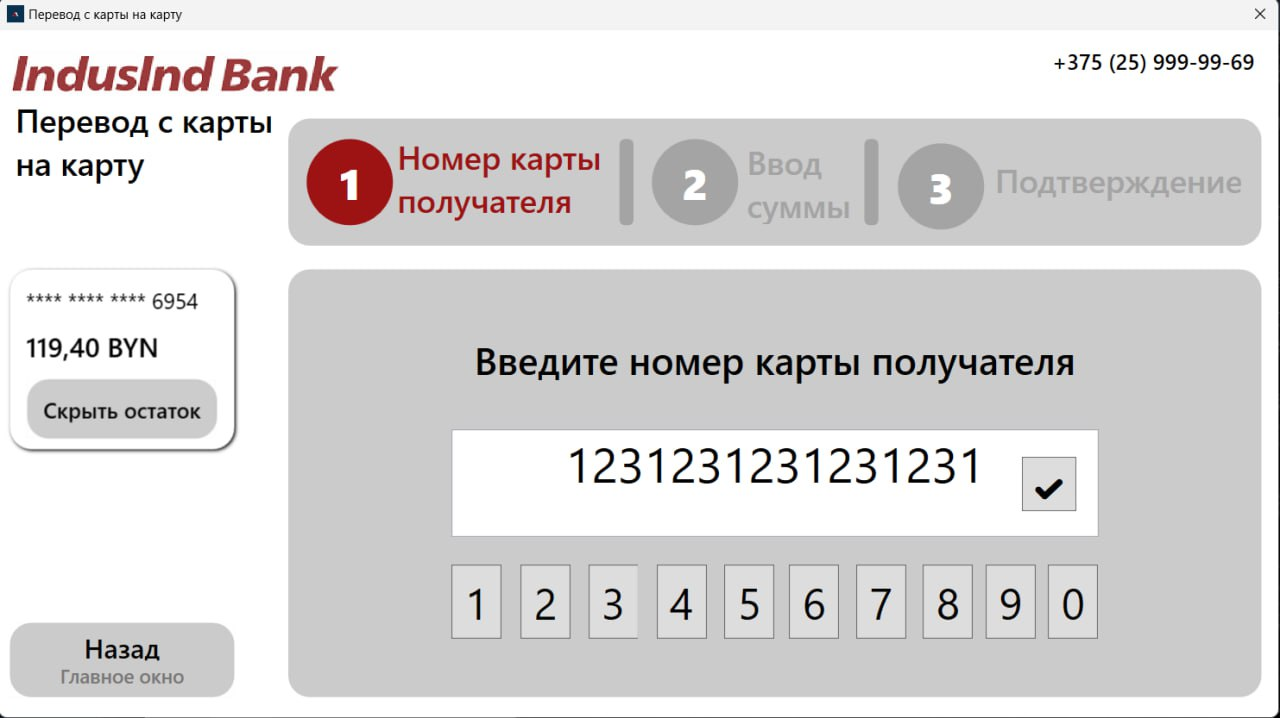


Рисунок 3.8 – Результат тест-кейс

Седьмой тест будет проведён на форме «Перевод с карты на карту».

Тест-кейс для тестирования функции перевод с карты на карту будет представлен в таблицы 3.9

Таблица 3.9 – Тест-кейс функции перевод с карты на карту

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 7 | Перевод на карту средств | 1. Укажите сумму перевода. 2. Выбрать валюту | Ожидаемый результат: переход на окно «подтверждение» |
| Фактический результат: успешный переход на окно «подтверждение» |

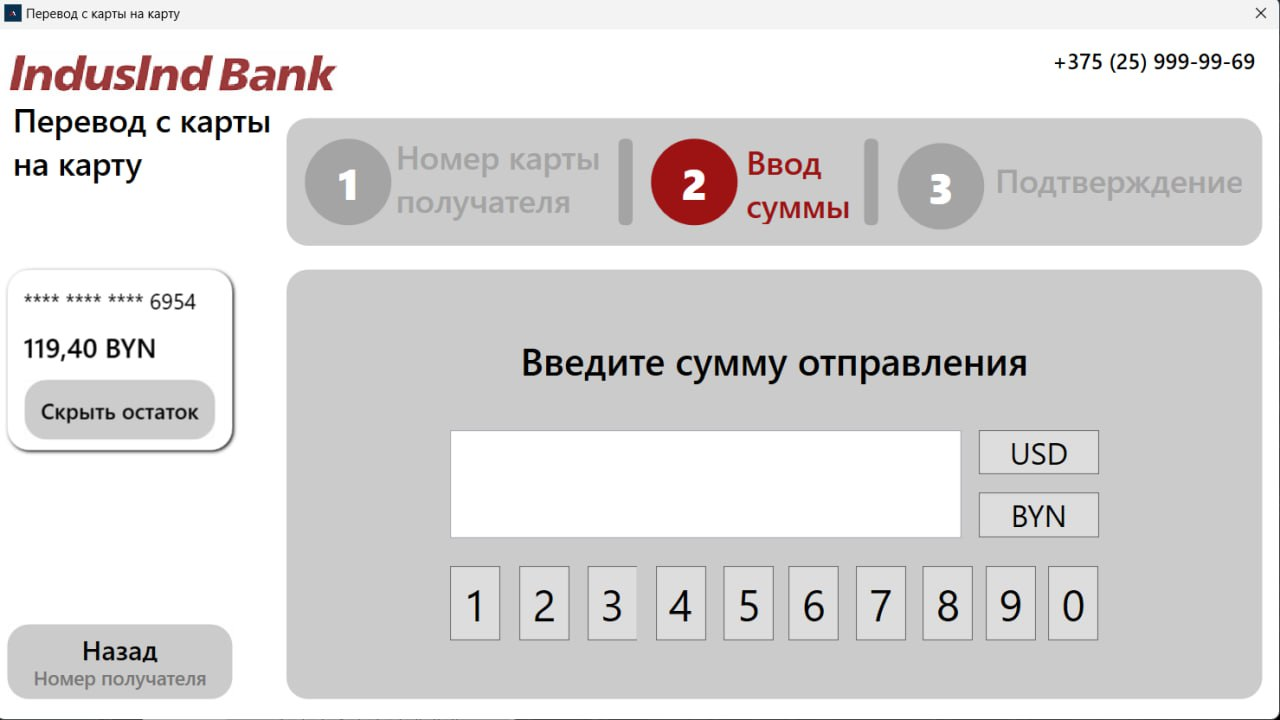


Рисунок 3.9 – Результат тест-кейс

Восьмой тест будет проведён на форме «Перевод с карты на карту».

Тест-кейс для тестирования функции перевод с карты на карту будет представлен в таблицы 3.10

Таблица 3.10 – Тест-кейс функции добавления перевод с карты на карту

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 8 | Перевод на карту средств | Нажмите кнопку ‘Подтвердить’ для инициации транзакции. | Ожидаемый результат: сообщение «перевод совершен» |
| Фактический результат: сообщение «перевод совершен» |

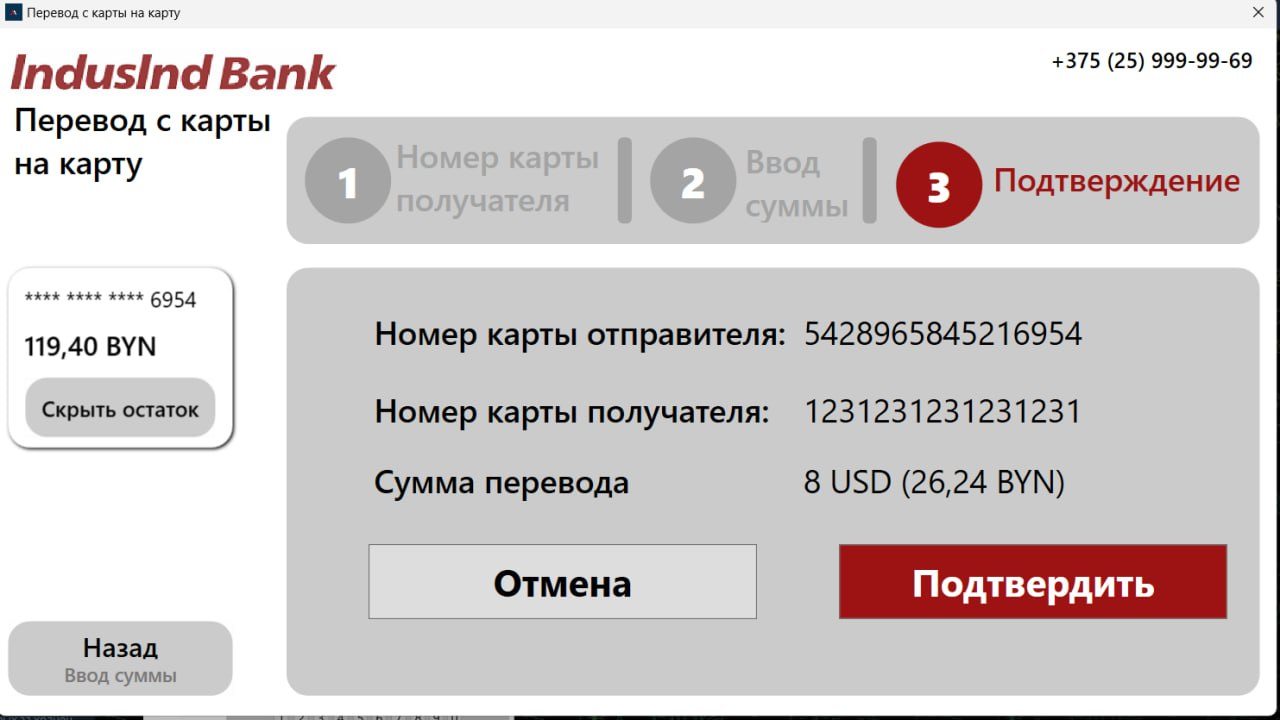


Рисунок 3.10 – Результат тест-кейс

Девятый тест будет проведён на форме «Приложения».

Тест-кейс для тестирования функции принятия заявок будет представлен в таблицы 3.11

Таблица 3.11 – Тест-кейс функции принятия заявок

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 9 | Рассмотрение заявления | 1. Просмотреть информацию 2. Нажать «принять» | Ожидаемый результат: сообщение «заявка принята» |
| Фактический результат: сообщение «заявка принята» |

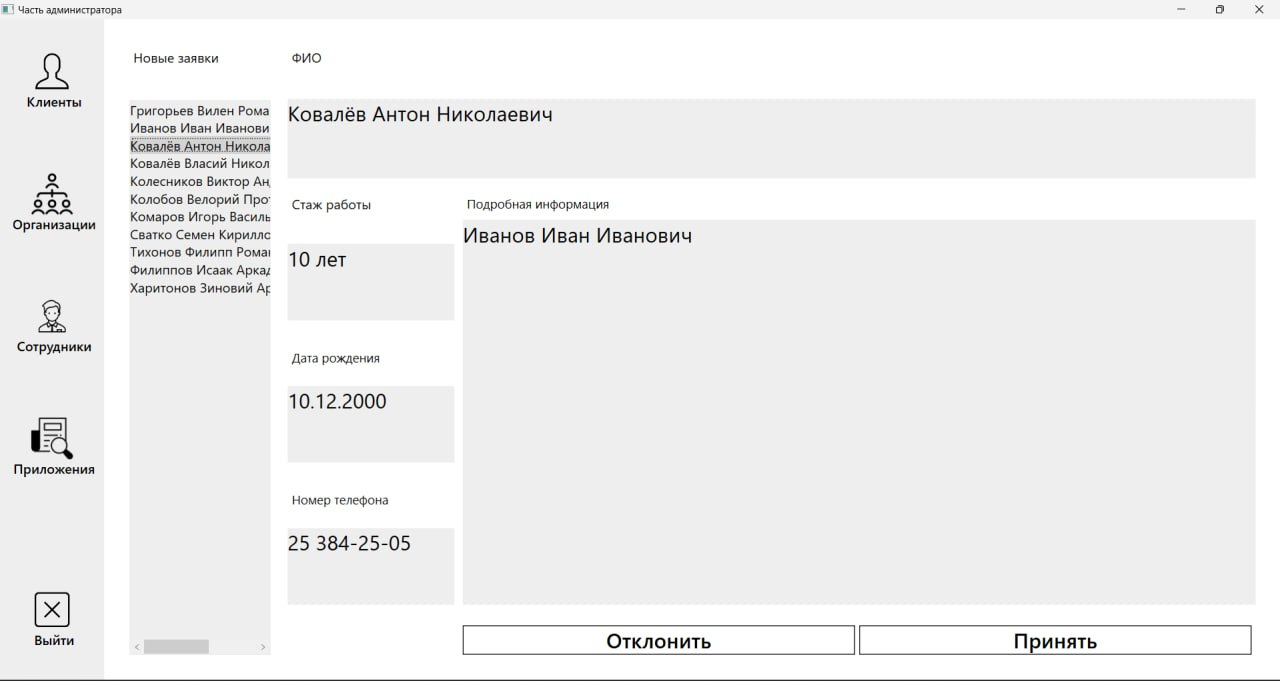


Рисунок 3.11 – Результат тест-кейс

Десятый тест будет проведён на форме «Список клиентов».

Тест-кейс для тестирования функции добавления клиента будет представлен в таблицы 3.12

Таблица 3.12 – Тест-кейс функции добавления клиентов

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 10 | Добавление клиента | 1. Нажать «добавить» 2. Заполнить соответствующие поля 3. Нажать «сохранить» 4. Нажать «обновить» | Ожидаемый результат: клиент добавлен |
| Фактический результат: клиент удачно добавлен |

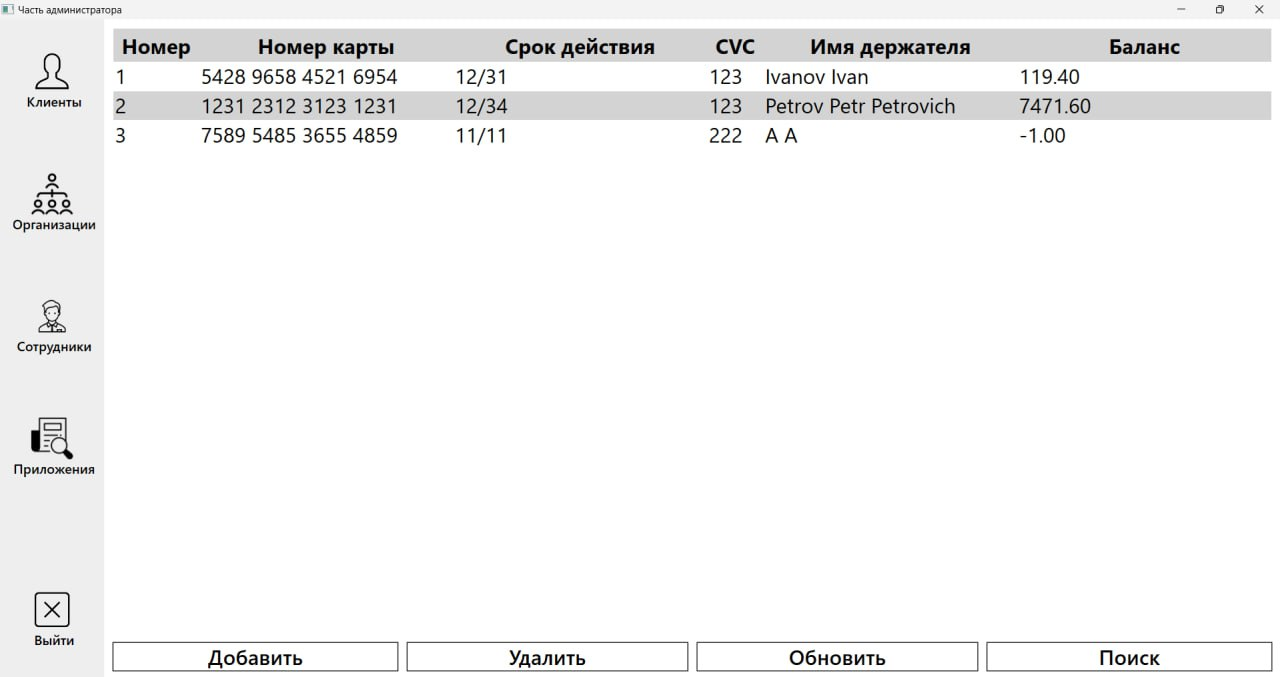


Рисунок 3.12 – Результат тест-кейс

Одиннадцатый тест будет проведён на форме «Список клиентов».

Тест-кейс для тестирования функции удаления клиента будет представлен в таблицы 3.13

Таблица 3.13 – Тест-кейс функции удаления клиентов

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 11 | удаление клиента | 1. Выбрать клиент 2. Нажать «удалить» | Ожидаемый результат: показ сообщения об удачном удалении клиента |
| Фактический результат: клиент удачно удален |

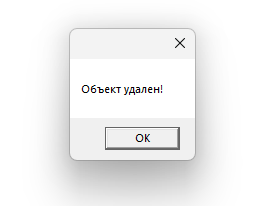


Рисунок 3.13 – Результат тест-кейс

Двенадцатый тест будет проведён на форме «Список сотрудников».

Тест-кейс для тестирования функции поиска клиента будет представлен в таблицы 3.14

Таблица 3.14 – Тест-кейс функции удаления клиентов

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-кейса | функция | Шаги выполнения | Результат |
| 12 | поиск | 1. Нажать «поиск» 2. Ввод имени сотрудника «Федор» 3. Нажать «найти» | Ожидаемый результат: вывод информации найденного сотрудника |
| Фактический результат: вывод информации найденного сотрудника |

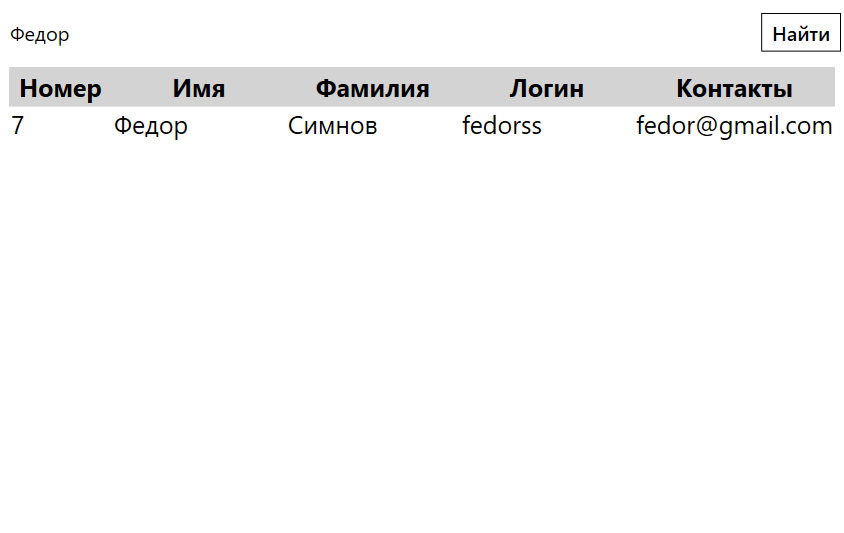


Рисунок 3.14 – Результат тест-кейс

# Описание справочной системы

Справочная система была разработана с использованием программного обеспечения Dr.Explain. Dr.Explain – это приложение, предназначенное для создания файлов справки, онлайн-руководств и документации.

Справочная система включает в себя несколько ключевых разделов, которые были определены в предоставленном изображении:

* авторизация пользователя;
* подача заявки на работу в нашем банке;
* снятие наличных;
* оплата платежей;
* перевод с карты на карту;
* настройки карты.

Справочная система может быть вызвана из главной формы при нажатии на кнопку «?».

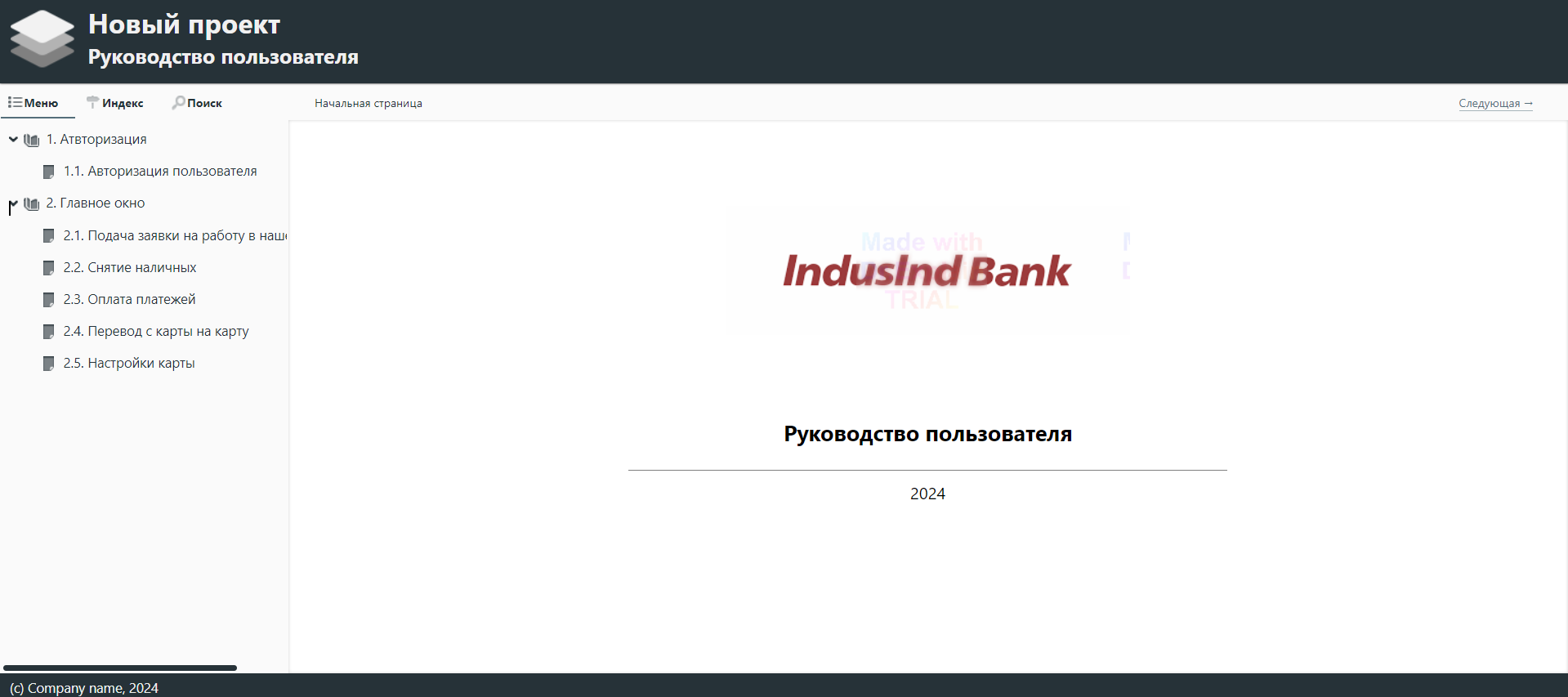


Рисунок 3.15 – Справочная система

# Применение

# Назначение программного средства

Приложение «Indusland Bank» предназначено для автоматизации банковских операций и оптимизации рабочего места сотрудников банка. Программное средство предоставляет администраторам возможности для поиска, добавления, отображения и удаления объектов в базе данных, таких как счета клиентов, счета организаций, сотрудники и принятые заявки. Оно также уменьшает затраты времени на выполнение рутинных банковских операций, ведет учет клиентов и позволяет быстро и удобно обрабатывать банковские заявки.

Основные разработанные функции:

* авторизация клиентов в системе;
* возможность перевода средств между счетами клиентов;
* возможность оплаты счета;
* возможность настройки персональные данные карты;
* возможность подачи заявки на работу в банке;
* возможность снятия наличных средств;
* возможность просмотра персональных данных.

Таким образом, «Indusland Bank» предлагает уникальный набор функций, которые помогут улучшить эффективность и производительность в области банковских операций, делая его важным инструментом для любого финансового учреждения, которое хочет оставаться конкурентоспособным в современном быстро меняющемся бизнес-окружении.

# Условия применения

Для применения данного программного средства необходимы следующие технические требования:

* процессор Intel Core i3 или выше;
* минимальный объём оперативной памяти 256 Мбайт;
* оперативная система Windows 10 и выше;
* платформа .Net Framework v4.8;
* видеокарта с объёмом памяти не менее 256 Мбайт;
* клавиатура, мышь.

Процесс установки программного средства «Indusland Bank.exe» включает следующие шаги:

* загрузка программы: скачайте установочный файл «Indusland Bank.exe» с github или носителя, предоставленного разработчиком;
* установка: запустите скачанный установочный файл и следуйте инструкциям мастера установки.

# Заключение

Проект «Indusland Bank» представляет собой программное средство, разработанное для автоматизации банковских операций и оптимизации рабочего места сотрудников банка. Организационная сущность задачи заключается в создании инструмента, который облегчит работу банковских служащих, менеджеров по работе с клиентами и администраторов. Предметная область проекта охватывает управление счетами клиентов, обработку транзакций, управление продуктами и услугами, а также генерацию чеков.

Технические требования для программы включают процессор Intel Core i3 или выше, минимальный объем оперативной памяти 100 Мбайт, операционную систему Windows 10 и выше, платформу .Net Framework v4.8 и видеокарту с объемом памяти не менее 256 Мбайт. Логическая организация данных включает описание структуры данных и связей между объектами (сотрудники, клиенты, организации, заявки). Физическая организация данных включает выбор СУБД и оптимизацию запросов. Справочная система описывает функциональность и предоставляет помощь пользователям.

Программное средство «Indusland Bank» предоставляет возможности поиска, добавления, отображения, редактирования и поиска объектов в базе данных. Входные данные включают информацию о клиентах, сотрудниках, организациях и заявках. Выходные данные могут быть чеками, добавления записей и другой информацией.

Среди преимуществ разработанного ПС можно выделить автоматическое управление транзакиями, гибкость и масштабируемость, а также интуитивный пользовательский интерфейс.

«Indusland Bank» представляет собой важный инструмент для банковских компаний, помогая им оставаться конкурентоспособными в современном бизнес-окружении.

# Список использованных источников

1. ГОСТ Р 54694-2011. Экономика. Термины и определения. - Введ. 2013-01-01. - М.: Экономлайф, 2012.
2. Что значит АРМ (Автоматизированное рабочее место)? [Электронный ресурс] – Режим доступа: (https://alexrovich.ru/info).
3. Обязанности менеджера по работе с клиентами – чем занимается и как [Электронный ресурс] – Режим доступа: https://sales-generator.ru/blog).
4. Разработка автоматизированного рабочего места менеджера [Электронный ресурс] – Режим доступа: https://spravochnick.ru/informatika).
5. Автоматизированное рабочее место (АРМ): что это? [Электронный ресурс] – Режим доступа: https://www.liga-commersantov.ru/articles).
6. Что такое автоматизированное рабочее место и зачем оно нужно? [Электронный ресурс] – Режим доступа: <https://rpa-robin.ru/blog/avtomatizaciya>).
7. Автоматизация рабочего места - RPA ROBIN [Электронный ресурс] – Режим доступа: https://www.atlassian.com/ru/agile).
8. 9 лучших программ для автоматизации рабочих процессов [2024 г.] [Электронный ресурс] – Режим доступа: https://nauchniestati.ru/spravka).
9. ТОП-26 программ мониторинга и контроля работы сотрудников за [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles).
10. Удобство и эффективность: разработка автоматизированного рабочего места [Электронный ресурс] – Режим доступа: https://appmaster.io/ru/blog).
11. Топ лучших приложений для управления персоналом для автоматизации [Электронный ресурс] – Режим доступа: https://www.guru99.com/ru/best).

# Приложение А

Текст программных модулей

private string type;

public infoPage(string type)

{

InitializeComponent();

this.type = type;

switch (type)

{

case "Employees":

{

// Создаем столбцы таблицы в коде

DataGridTextColumn column1 = new DataGridTextColumn();

column1.Header = "Номер";

column1.Binding = new Binding("ID");

column1.Width = new DataGridLength(1, DataGridLengthUnitType.Auto);

DataGridTextColumn column2 = new DataGridTextColumn();

column2.Header = "Имя";

column2.Binding = new Binding("FirstName");

column2.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column5 = new DataGridTextColumn();

column5.Header = "Фамилия";

column5.Binding = new Binding("LastName");

column5.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column3 = new DataGridTextColumn();

column3.Header = "Логин";

column3.Binding = new Binding("Login");

column3.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column4 = new DataGridTextColumn();

column4.Header = "Контакты";

column4.Binding = new Binding("Contact");

column4.Width = new DataGridLength(1, DataGridLengthUnitType.Auto);

// Добавляем созданные столбцы в таблицу

dataGrid.Columns.Add(column1);

dataGrid.Columns.Add(column2);

dataGrid.Columns.Add(column5);

dataGrid.Columns.Add(column3);

dataGrid.Columns.Add(column4);

dataGrid.ItemsSource = ClientHelper.getAllAdmins();

break;

}

case "Clients":

{

// Создаем столбцы таблицы в коде

DataGridTextColumn column1 = new DataGridTextColumn();

column1.Header = "Номер";

column1.Binding = new Binding("ID");

column1.Width = new DataGridLength(1, DataGridLengthUnitType.Auto);

DataGridTextColumn column2 = new DataGridTextColumn();

column2.Header = "Номер карты";

column2.Binding = new Binding("number");

column2.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column3 = new DataGridTextColumn();

column3.Header = "Срок действия";

column3.Binding = new Binding("dateCard");

column3.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column4 = new DataGridTextColumn();

column4.Header = "СVC";

column4.Binding = new Binding("cvc");

column4.Width = new DataGridLength(1, DataGridLengthUnitType.Auto);

DataGridTextColumn column5 = new DataGridTextColumn();

column5.Header = "Имя держателя";

column5.Binding = new Binding("ownerName");

column5.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column6 = new DataGridTextColumn();

column6.Header = "Баланс";

column6.Binding = new Binding("balance");

column6.Binding.StringFormat = "{0:0.00}";

column6.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column7 = new DataGridTextColumn();

column7.Header = "Логин";

column7.Binding = new Binding("Login");

column7.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

dataGrid.Columns.Add(column1);

dataGrid.Columns.Add(column2);

dataGrid.Columns.Add(column3);

dataGrid.Columns.Add(column4);

dataGrid.Columns.Add(column5);

dataGrid.Columns.Add(column6);

this.dataGrid.ItemsSource = ClientHelper.getAllClients();

break;

}

case "Organisations":

{

// Создаем столбцы таблицы в коде

DataGridTextColumn column1 = new DataGridTextColumn();

column1.Header = "Номер";

column1.Binding = new Binding("ID");

column1.Width = new DataGridLength(1, DataGridLengthUnitType.Auto);

DataGridTextColumn column2 = new DataGridTextColumn();

column2.Header = "Название";

column2.Binding = new Binding("organisationName");

column2.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column3 = new DataGridTextColumn();

column3.Header = "Адрес";

column3.Binding = new Binding("organizationAdress");

column3.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

DataGridTextColumn column6 = new DataGridTextColumn();

column6.Header = "Баланс";

column6.Binding = new Binding("balance");

column6.Binding.StringFormat = "{0:0.00}";

column6.Width = new DataGridLength(1, DataGridLengthUnitType.Star);

dataGrid.Columns.Add(column1);

dataGrid.Columns.Add(column2);

dataGrid.Columns.Add(column3);

dataGrid.Columns.Add(column6);

dataGrid.ItemsSource = ClientHelper.getAllOrganisations();

break;

}

}

}

private void Button\_Click(object sender, RoutedEventArgs e)//ДОБАВИТЬ

{

try

{

switch (type)

{

case "Employees":

{

addAdminForm form = new addAdminForm();

form.ShowDialog();

break;

}

case "Clients":

{

addClientForm form = new addClientForm();

form.ShowDialog();

break;

}

case "Organisations":

{

addOrganisationForm form = new addOrganisationForm();

form.ShowDialog();

break;

}

}

}

catch (Exception ex) { MessageBox.Show(ex.ToString()); }

}

private void Button\_Click\_1(object sender, RoutedEventArgs e) // удаление

{

try

{

var selected = (dynamic)dataGrid.SelectedItem;

switch (type)

{

case "Employees":

{

ClientHelper.deleteAdministrators(selected.ID);

break;

}

case "Clients":

{

ClientHelper.deleteClient(selected.ID);

break;

}

case "Organisations":

{

ClientHelper.deleteOrganisation(selected.ID);

break;

}

}

MessageBox.Show("Объект удален!");

NavigationService.Navigate(new infoPage(type));

}

catch (Exception ex) { MessageBox.Show(ex.ToString()); }

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new infoPage(type));

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

searchForm form = new searchForm(type);

form.ShowDialog();

}

# Приложение Б

Выходные документы

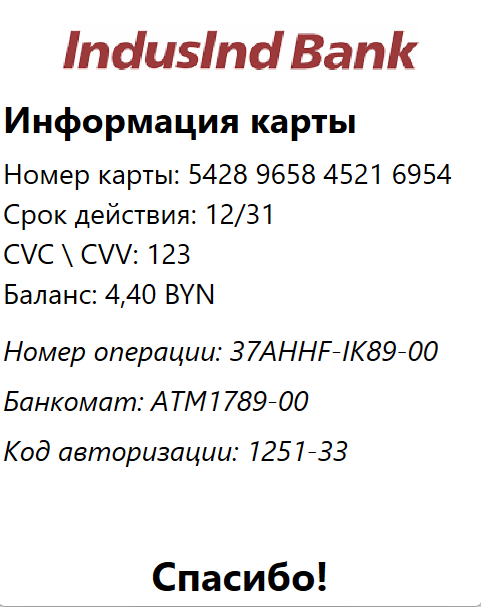


Рисунок 3.1 – Информация персональной карты

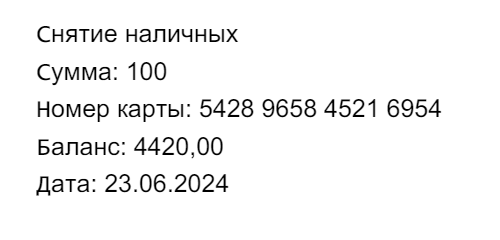


Рисунок 3.2 – Чек снятия наличных со счета