# Глава №1.

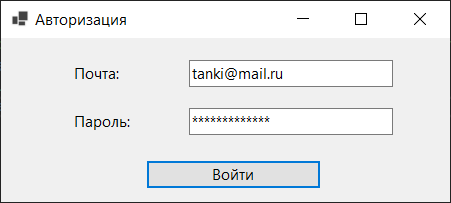
# ЭТАП №1. ФОРМУЛИРОВКА ТЗ

## Техническое задание на разработку программного продукта «Банк «Вы банкрот». Исполнитель»

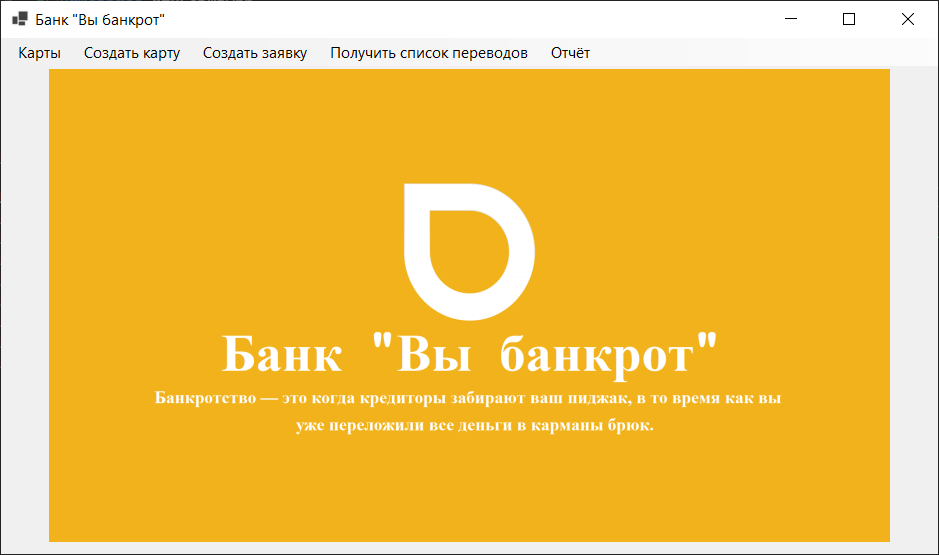
## Программа «Банк «Вы банкрот». Клиент» предназначена для ведения учета заявок о пополнения и снятия средств с карт клиента. Клиент может создавать свои карты и создавать заявки на пополнение и снятие средств. Программа предоставляет возможность получения отчетов по заявкам с определенной карты.

## Основной функционал программы:

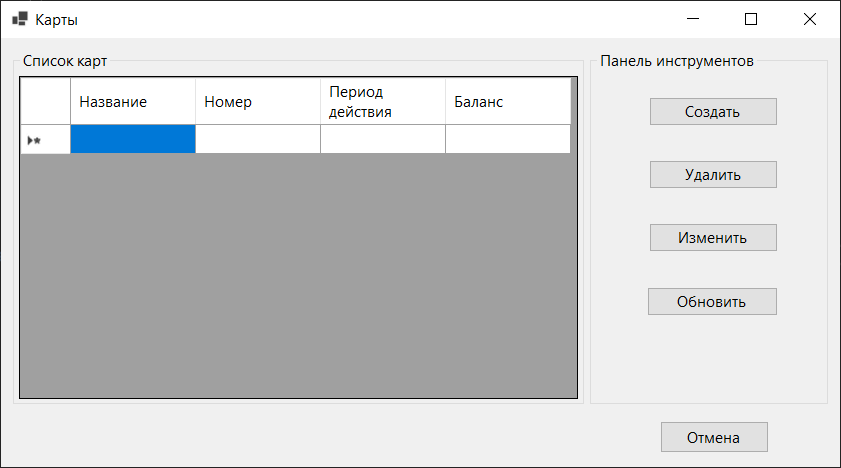
* Авторизация. Пользователь получает сразу готовый аккаунт с нужным логином (почтой) и паролем. Пользователь вводит их и, после проверки, происходит переход к основной форме. Если такого проверка прошла неуспешно, выводится сообщение с просьбой повторного ввода.



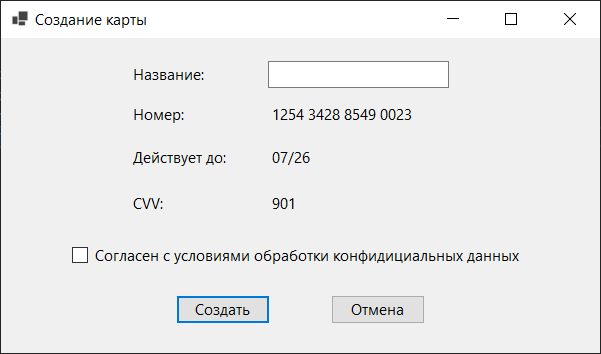
* Основная форма. Состоит из меню и логотипа. Через пункты меню можно перейти на форму создания карты, просмотреть собственные карты, создать заявку на пополнение или снятие средств, получить общий отчет по операциям или только по нескольким картам.



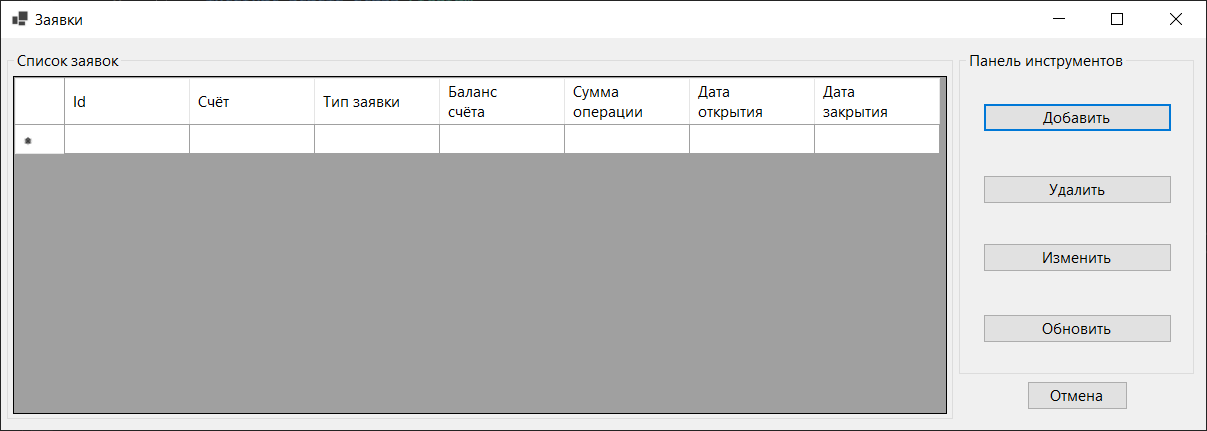
* Формирование карт (CRUD). - Имеется форма со списком всех действующих карт клиента и кнопки для создания, изменения и удаления карты.



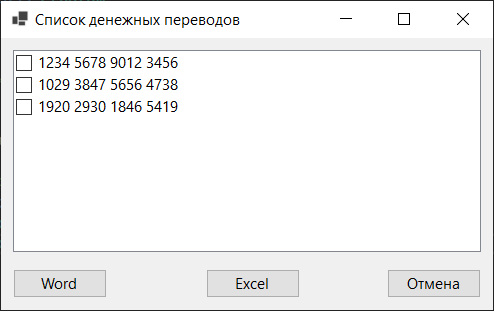
* Создание и редактирование карты. Здесь пользователь может создать свою карту. Также для создания карты нужно согласиться с политикой компании. Есть возможность дать название карте, если пользователь это захочет.



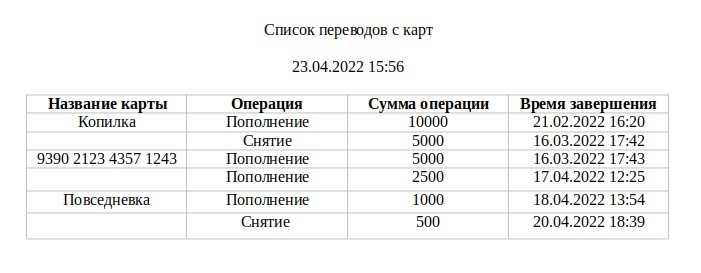
* Создать заявку. Здесь пользователь может создать и отправить заявку на обработку кассиром. Пользователю нужно выбрать свою карту из списка, выбрать тип операции и ввести сумму операции, а затем нажать кнопку создать.



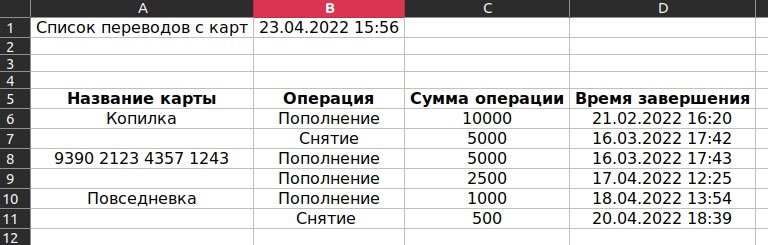
* Получить список переводов. Здесь пользователь может получить отчет по операциям с определенных карт. Пользователь выбирает нужные карты, и после того выбирает формат сохранения.



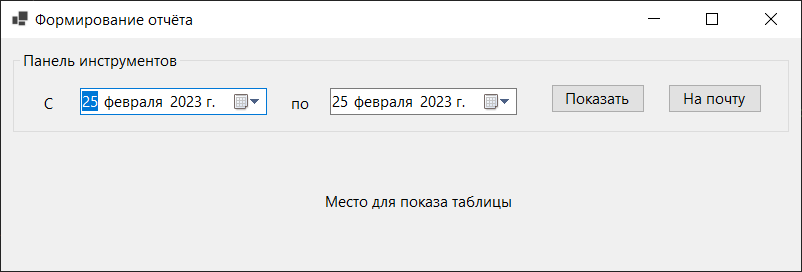
Пример сохранения в формате Word:



Пример сохранения в формате Excel:



* Отчет. Здесь у пользователя есть возможность получить общий отчет. Пользователю нужно выбрать даты. Затем пользователь может просмотреть отчет. Также может отправить на почту в формате PDF.



В качестве интерфейса пользователя будет выступать Web-клиент,   
разработанный на ASP.NET.

## Сравнение аналогов

## В качестве аналогов были рассмотрены следующие программные продукты:

1. Тинькофф:

Один из самых известных частных банков России. Его мобильное приложение является самым лучшим онлайн банком на данный момент.

Преимущества: отличный дизайн, скорость обработки действий пользователя, стабильность работы приложения, большое количество программ кэшбэка.

Недостатки: перегруженность интерфейса, большое количество ненужных опций, большое количество рекламы.

1. Сбербанк-онлайн:

Приложение Сбербанка – приложение самого популярного банка России. Его мобильное приложение является наиболее распространённым онлайн банком.

Преимущества: стабильность работы приложения, приветливый интерфейс, некоторые возможности мессенджера, наличие бонусов «СберСпасибо».

Недостатки: перегруженность интерфейса, периодическое зависание при обработке денежных переводов, большое количество рекламы.

1. Открытие:

Преимущества: понятность интерфейса, простота отчётов по операциям,

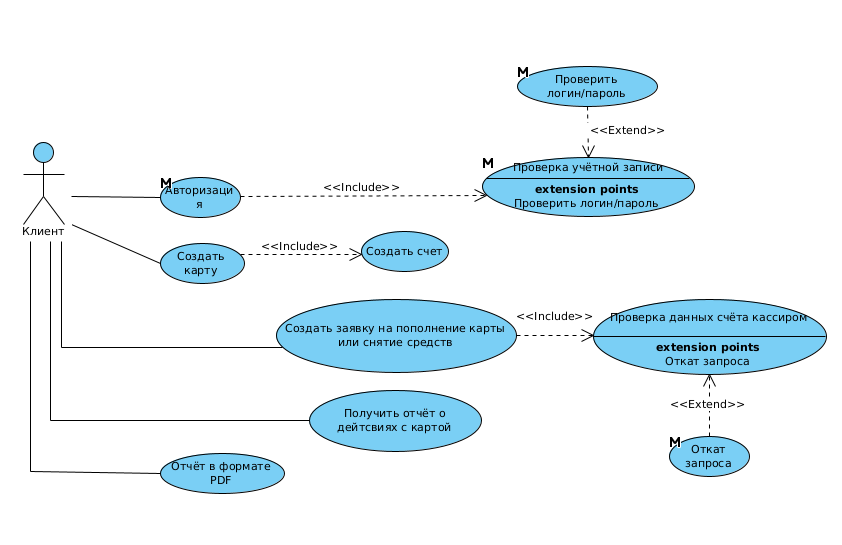
Недостатки: «рубленый дизайн», порой чрезмерная проста интерфейса выглядит как минус, частые большие (по размеру) обновления.

# ЭТАП №2. СОСТАВЛЕНИЕ ДИАГРАММ

## Диаграмма вариантов использования

## Диаграмма вариантов использования отражает отношения между актёрами и прецедентами и является составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

## Диаграмма вариантов использования исполнителя:

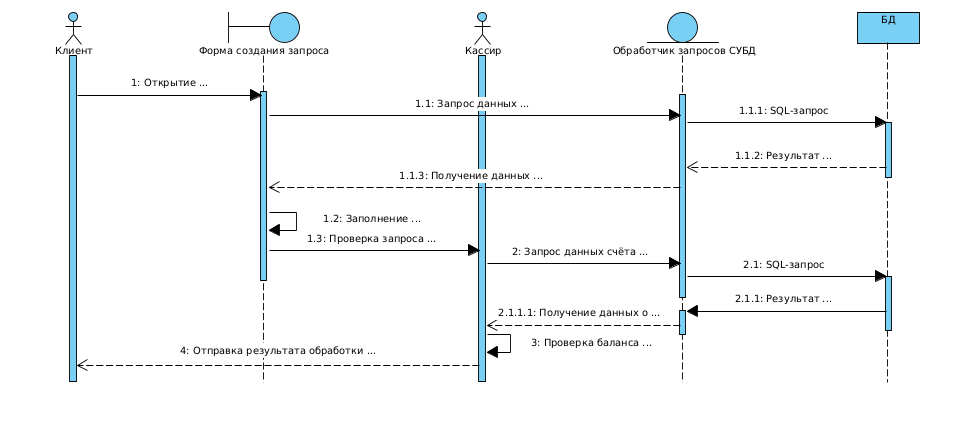


Клиент может авторизоваться, при этом, идет проверка на ошибки. Клиент также может создать карту. Вместе с новой картой создастся счет. Клиент может отправить запрос кассиру на операцию с карты. При ошибке кассир отменит запрос. Также клиент может получить отчеты в виде файла, либо отчет отправится на почту

## Диаграмма последовательности

## Это диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов)

**Диаграмма последовательности исполнителя:**

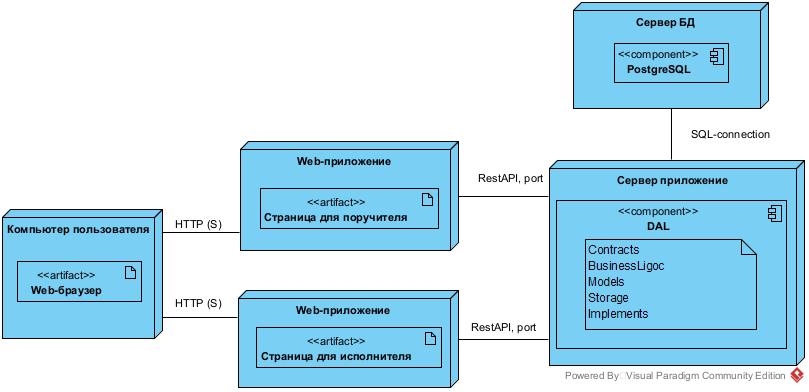


Рассмотрим этапы формирования запроса:

Клиент открывают форму. Форма запрашивает данные о картах и возможных операциях. Клиент дополняет форму и отправляет кассиру. И уже кассир принимает или отклоняет заявку.

## Диаграмма развертывания

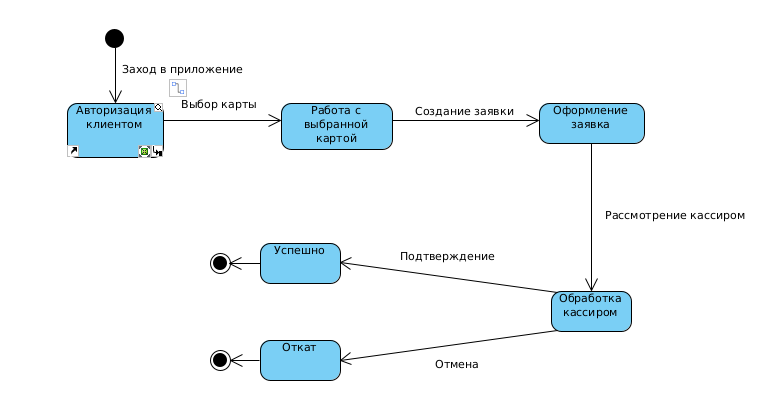
Диаграмма развёртывания в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI)



## Диаграмма состояний

Диаграмма состояний — это, по существу, ориентированный граф для конечного автомата из теории автоматов со стандартизированными условными обозначениями, в котором вершины обозначают состояния, а дуги показывают переходы между двумя состояниями. Диаграмма может определять множество систем от компьютерных программ до бизнес-процессов. Используются следующие условные обозначения:

**Диаграмма состояний исполнителя:**



Рассмотрим этапы формирования запроса

Клиент открывают форму. Форма запрашивает данные о картах и возможных операциях. Клиент дополняет форму и отправляет кассиру. И уже кассир принимает или отклоняет заявку.

# ЭТАП №3. СОСТАВЛЕНИЕ СХЕМЫ БД.

Основная цель – составление схемы базы данных, какие таблицы будут в БД, какие в них будут поля и как таблицы будут связаны друг с другом.

Нормализация:

1. Клиент — исполнитель, имеет атрибуты описывающие себя.
   1. ID
   2. Имя
   3. Фамилия
   4. Отчество
   5. Телефон
   6. Почта (Уникальный)
   7. Пароль
2. Карта — карта пользователя. У одного пользователя может быть много карт, поэтому карта связана один ко многим с клиентом. Также связан со счетом в банке (будет описан ниже) один к одному (По общему решению, семейных карт мы не делаем, одна карта - один счет)
   1. ID
   2. Имя (может быть пустым)
   3. ID клиента
   4. ID счета
   5. Номер карты (является уникальным идентификатором, однако из-за сложной структуры было решено не делать его первичным ключом)
   6. Период
   7. CVV
3. Банковский счет — сам счет в банке, с ним работает только кассир.
   1. ID
   2. Баланс
   3. Дата открытия
4. Тип заявки — снятие / перевод денег на карту.
   1. ID
   2. Название (Уникальный)
5. Заявка — заявка клиента, отправляемая кассиру. Содержит связь с картой (с которой производится операция) и кассиром (принимающим запрос).
   1. ID
   2. ID типа заявки
   3. ID карты
   4. Сумма операции
   5. ID Кассира
   6. Время создания
   7. Время завершения
6. Кассир — сущность поручителя имеет атрибуты, которые описывают себя
   1. ID
   2. Имя
   3. Фамилия
   4. Отчество
   5. Телефон
   6. Почта (Уникальный)
   7. Пароль
7. Тип операции — снятие/перевод денег со счета
   1. ID
   2. Название (Уникальный)
8. Операции со счетом — сущность для записи кассиром операций, производимых со счётом. Хранит информацию о типе операции.
   1. ID
   2. ID Счета
   3. Сумма операции
   4. ID Типа операции
   5. Время выполнения

## ER-Диаграмма

# ЭТАП №4. ПРЕДСТАВЛЕНИЕ ПРОЕКТОВ

На данной стадии выполнено:

* описание предметной области
* рассмотрение аналогов
* диаграммы, описывающие поведение системы
* диаграмма базы данных

Осталось рассмотреть применяемые технологии, а также распределение обязанностей:

**Применяемые технологии:**

1. Web-клиент на ASP.NET

**Распределение обязанностей:**

1. FRONTEND, Проектирование логики страниц, тестирование – Елисеев Егор
2. BACKEND, логика генерации отчётов, проектирование БД – Захарченко Марк.