# Глава №1.

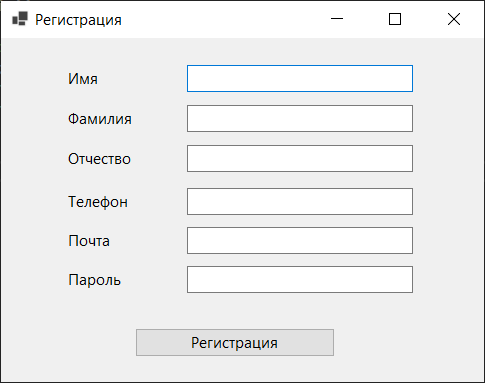
# ЭТАП №1. ФОРМУЛИРОВКА ТЗ

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

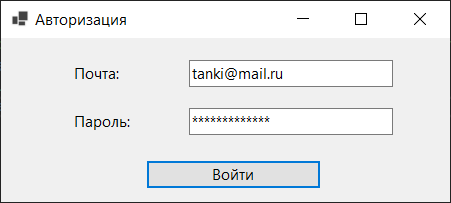
Программа «Банк «Вы банкрот». Поручитель» предназначена для ведения записей кассира о выдаче наличных средств со счёта клиенту, а также о произведённых клиентом денежных переводах. Кассиры могу открывать счета, записи о выдаче с них наличных клиентам, а также о произведённых с них переводах. Программа предоставляет возможность получения отчетов за период по выдаче наличных и переводам денег, указываемым в записях ко всем счетам клиентов.

Основной функционал программы:

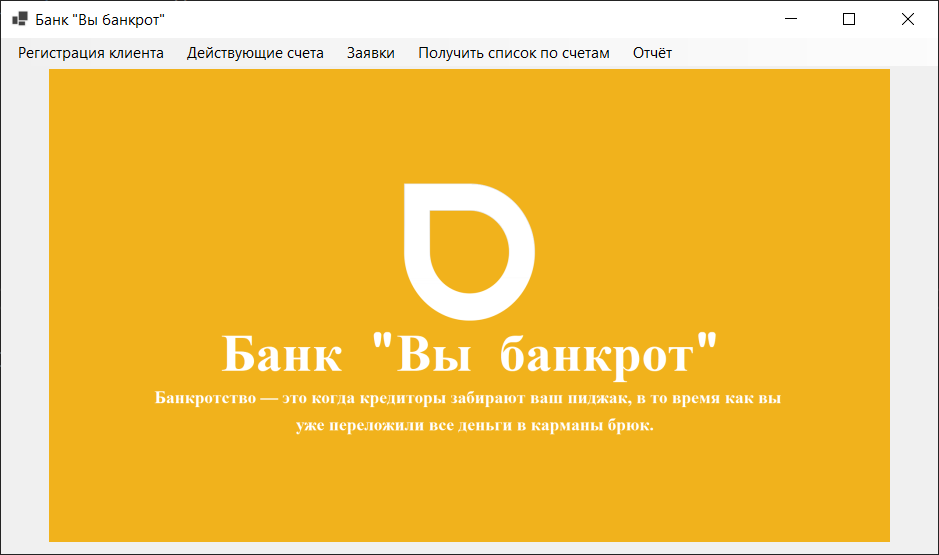
* Регистрация пользователя. Для регистрации пользователя кассир должен заполнить следующие данные:
  + ФИО (раздельно).
  + Телефон
  + Почта.
  + Пароль.



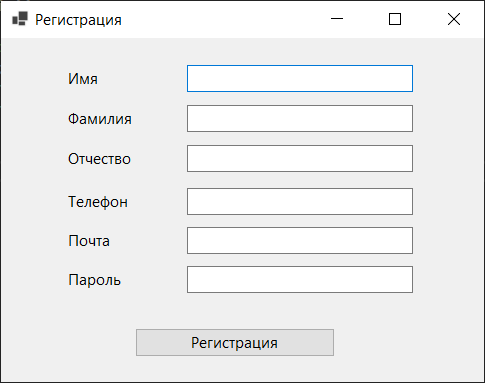
* Авторизация. Пользователь вводит почту и пароль. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



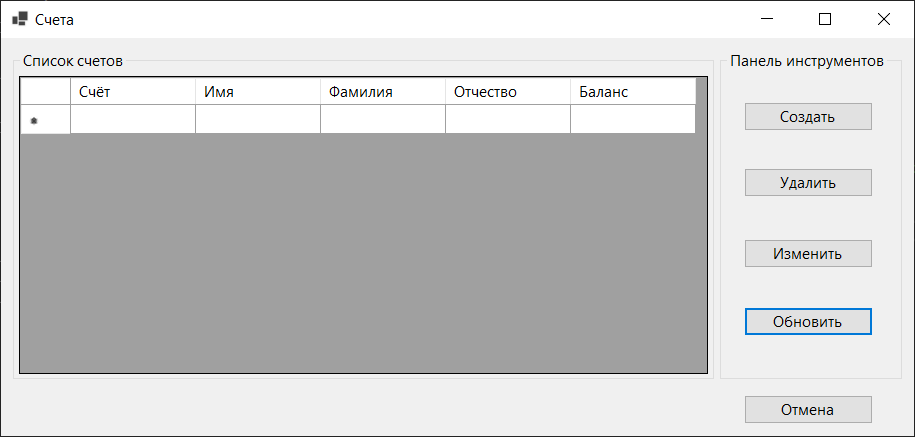
* Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы для работы со счетами, выдачей наличных денег, операциями на пополнение карты (в том числе и привязка к переводу денег), на форму для получения списка заявок и на форму для получения отчета по выдаче наличных и переводам денег. Также здесь кассир может перейти в меню регистрации клиента



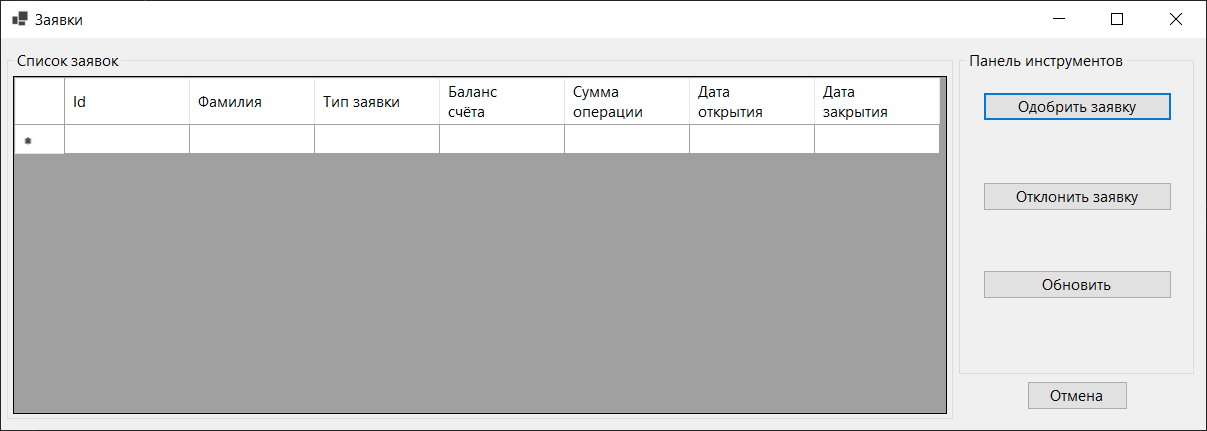
* Регистрация. Для регистрации клиента, кассир должен заполнить следующие данные:
  + ФИО (раздельно).
  + Телефон
  + Почта.
  + Пароль.



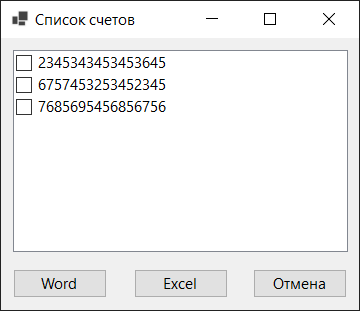
* Формирование счетов (CRUD). Имеется форма со списком всех действующих счетов банка и кнопки для создания, изменения и удаления счета



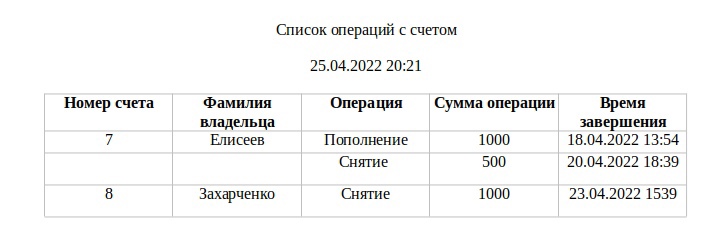
* Формирование заявок — Здесь хранятся данные о заявках клиента (Или клиента). Здесь можно просмотреть определенную заявку, одобрить или отклонить ее. Здесь хранится ФИО клиента, номер карты, название операции, сумма операции, статус, дата создания, дата закрытия.



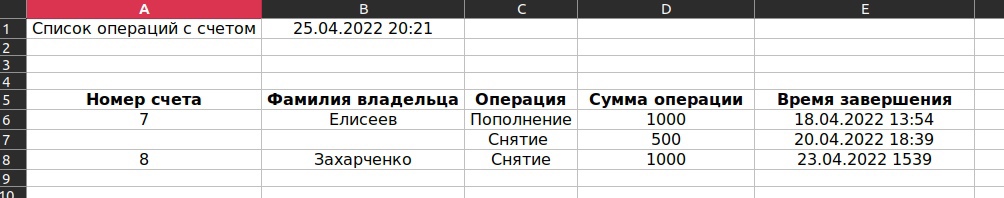
* Получить список по счетам. Здесь пользователь может получить отчет по операциям с определенных счетов. Пользователь выбирает нужные счета, и после этого выбирает формат сохранения.

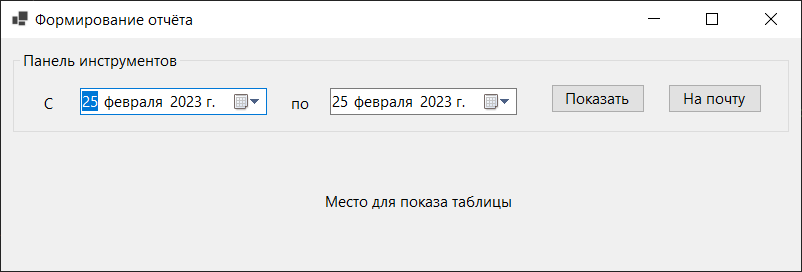


Пример сохранения в формате Word:



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,

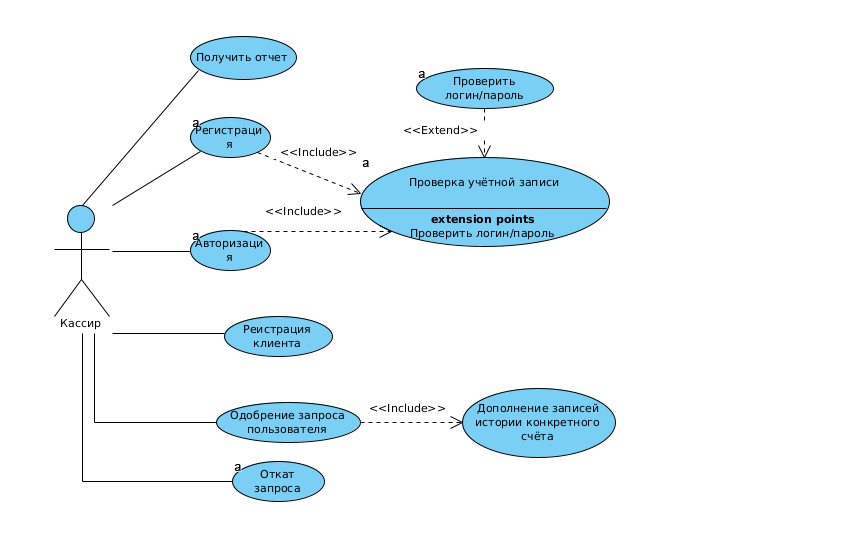
Недостатки: «рубленый дизайн», порой чрезмерная проста интерфейса выглядит как минус, частые большие (по размеру) обновления.

# ЭТАП №2. СОСТАВЛЕНИЕ ДИАГРАММ

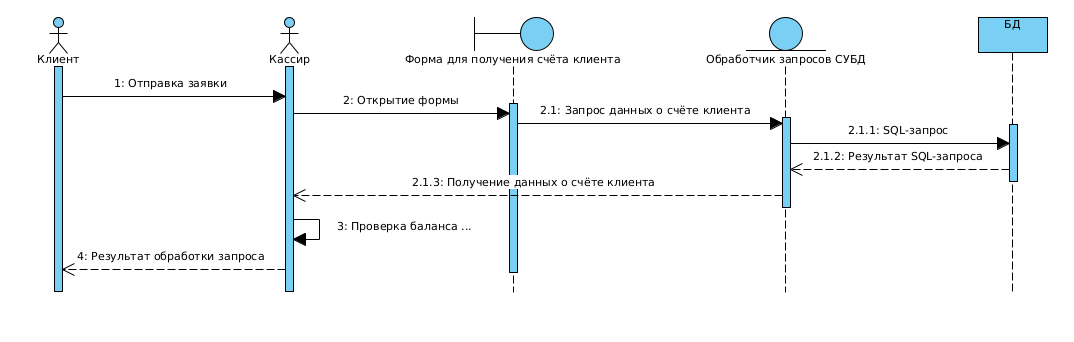
## Диаграмма вариантов использования

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

## Диаграмма вариантов использования поручителя:

 Кассир может зарегистрироваться и войти в аккаунт, присутствует проверка логина и пароля. Кроме этого, кассир может зарегистрировать клиента. Кассир имеет право принимать и отклонять запросы пользователей. При этом он фиксирует все изменение в счете. Также кассир может получить отчеты в виде файла, либо отчет отправится на почту

### Диаграмма последовательности поручителя.

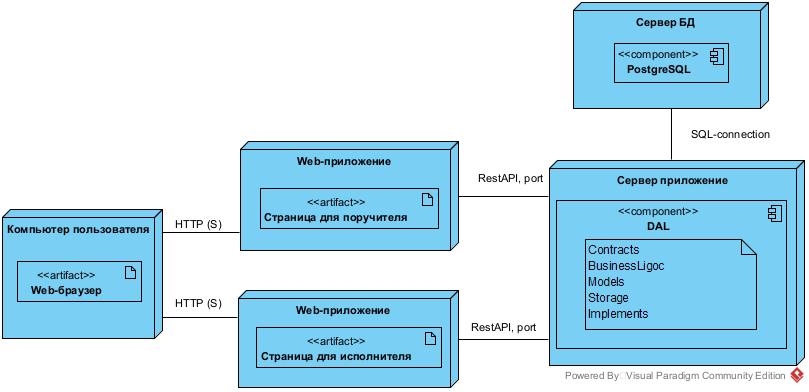


Рассмотрим ситуацию обработку запроса клиента кассиром.

Кассиру приходит заявка. Кассир переходит в форму обработки запроса. Форма запрашивает данные о счете клиента (определенной карты). После кассир проверяет счет и после решает принять или отклонить 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)

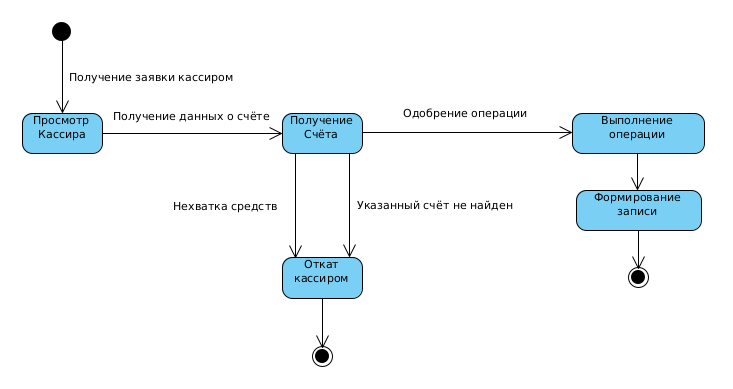


## Диаграмма состояний

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 поручителя.**

Рассмотрим ситуацию обработку запроса клиента кассиром.



тавление схемы базы данных, каирены следующие прграпрапрапрапрапрапрапрапапрапКассиру приходит заявка. Кассир переходит в форму обработки запроса. Форма запрашивает данные о счете клиента (определенной карты). После кассир проверяет счет и после решает принять или отклонить 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.