МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра обработки информации и машинного обучения

Курсовая работа по курсу

«Технологии программирования»

Веб-приложение «Online\_bank»

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_А.В. Калинина,3 курс, д/о

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_ Т.В. Акиндинова,3 курс, д/о

Руководитель \_\_\_\_\_\_\_\_\_\_\_ М.С. Мишанин

Воронеж 2019

**ОГЛАВЛЕНИЕ**

[Введение 3](#_Toc10518343)

[Глоссарий 4](#_Toc10518344)

[1. Постановка задачи 5](#_Toc10518345)

[2. Анализ предметной области 7](#_Toc10518346)

[2.1 Существующие решения 8](#_Toc10518347)

[2.2 IDEF0 10](#_Toc10518348)

[2.3 Диаграмма прецедентов 12](#_Toc10518349)

[3. Проектирование приложения 14](#_Toc10518350)

[3.1 ER-диаграмма 14](#_Toc10518351)

[3.2 Диаграмма состояний 16](#_Toc10518352)

[3.3 Диаграмма активности 16](#_Toc10518353)

[3.4 Диаграммы последовательностей 16](#_Toc10518354)

[3.5 Диаграммы коммуникации 20](#_Toc10518355)

[3.6 Диаграмма развёртывания 24](#_Toc10518356)

[3.7 Диаграмма классов 25](#_Toc10518357)

[3.8 Диаграмма объектов 26](#_Toc10518358)

[4. Реализация приложения 27](#_Toc10518359)

[4.1 Средства реализации 27](#_Toc10518360)

[5. Планирование работ 29](#_Toc10518361)

[5.1 Виды работ в процессе разработки 29](#_Toc10518362)

[Заключение 31](#_Toc10518363)

[Список литературы 32](#_Toc10518364)

# Введение

Все мы живём в обществе, которое всё время что-то покупает и продаёт. На этом обмене строится вся экономическая деятельность людей. Поэтому трудно представить наш мир без банковской системы. Сейчас банки предоставляют нам широкий спектр услуг, значительно упрощающих нашу хозяйственную деятельность.

Так как общество часто сталкиваются с этой сферой жизни, то необходимо было автоматизировать её, чтобы люди быстро и легко производили всевозможные действия с своими финансами. Для этого придумали приложения для различных технических устройств, которые уже прочно обосновались в нашей жизни.

Теперь обладатель такого приложения может, не выходя из дома, осуществлять различные покупки, оплачивать счета, заказывать билеты и тд. всего за пару минут. Также он сможет в любой момент времени узнать свой баланс на любом счёте и ему не надо будет идти ради этого в банк или пересчитывать непосредственно все денежные средства. Такой комфорт и быстродействие обеспечивает широкую распространённость и актуальность данного приложения.

Данная курсовая работа посвящена разработке банковского веб-приложения, которое позволит пользователям взаимодействовать с банком, заходя в свой личный кабинет, и производить различные операции с деньгами, а также видеть всю информацию о своих счетах и предложениях банка.

# Глоссарий

**Незарегистрированный пользователь** – человек, который заходит на сайт и может просмотреть основную информацию о банке и его предложениях, но не имеет доступ в личный кабинет.

**Неавторизованный пользователь** –человек, который имеет доступ в личный кабинет, но в данный момент не прошёл авторизацию

**Пользователь** –человек, который имеет доступ в личный кабинет

**Клиент** –человек, который имеет через свой личный кабинет имеет доступ к клиентским возможностям системы

**Оператор** – человек, который работает в банке и имеет доступ через свой личный кабинет к особым возможностям веб-приложения (например, добавление/удаление предложений).

**Предложение** – услуга, которую предоставляет банк своим пользователям.

**Вклад** – денежные средства, которые передаёт в финансовое учреждение физическое или юридическое лицо с целью получения прибыли в виде процентов.

**Кредит** – денежные средства, которые получает физическое или юридическое лицо на счёт под определённые проценты и на определённый срок.

**Дебет** – счёт для хранения и распоряжения собственными денежными средствами.

**Транзакция** – любая единичная операция с использованием счёта.

# Постановка задачи

**Цель:** разработать веб-приложение банка для удобного взаимодействия клиента с его счетами. Оно должно представлять действующие предложения Банка, знакомить с ними пользователя, также позволять зарегистрироваться новому клиенту банка в системе.

**Сфера использования:**

Данная система направлена на пользование:

* Клиенты банка;
* Оператор банка;
* Любым человеком, который хочет узнать информацию о банке или стать его клиентом

**Требования:**

Планируется реализовать:

* Регистрацию новых пользователей банка;
* Ведение учетных записей клиентов (т.е. поддержка личных кабинетов);
* Предоставление клиентам данных об их счетах/ вкладах/кредитах;
* Предоставление возможности клиентам выполнять операции по всем своим счетам/ вкладам/кредитам;
* Предоставление возможности как клиентам, так и не зарегистрированным пользователям просматривать предложения банка по предоставляемым услугам.

**Задачи:**

1. Провести анализ требований к разрабатываемой системе.
2. Провести проектирование приложения.
3. Реализовать веб-приложение, удовлетворяющее указанным требованиям, и описать процесс разработки и итоговый результат.

# Анализ предметной области

Необходимо разработать веб-приложение для взаимодействия пользователя с предоставляемыми услугами банка, т.е. операциями со счетами, просмотр предложений и информации о своих счетах. Чтобы это сделать, нужно понять какие сущности и с какими параметрами у нас есть в системе.

Прежде всего, у нас есть клиенты, их параметры:

* ФИО
* Серия и номер паспорта
* Логин
* Пароль
* Дата рождения

Также у нас есть операторы с параметрами:

* Имя
* Логин
* Пароль

У каждого клиента банка есть хотя бы один счёт, для хранения информации о нём необходима ещё одна сущность со следующими параметрами:

* Сумма
* Дата открытия
* Дата закрытия

Счета клиентов могут быть трёх типов: дебетовый, кредитный, депозитный. Эту информацию также необходимо хранить в системе.

При открытии вклада или взятие кредита, клиенту необходимо просмотреть существующие предложениях банка и выбрать одно из них. Предложение банка будут характеризоваться следующими параметрами:

* Название
* Процентная ставка
* Тип(дебет/кредит/вклад)
* Период

Так как в системе предполагается перевод денежных средств между счетами, то необходима ещё одна сущность транзакций с параметрами:

* Счёт списания
* Счёт зачисления
* Сумма
* Дата

Для всех этих сущностей будут созданы таблицы в базе данных.

## Существующие решения

**Сбербанк**

Приложение Сбербанк позволяет клиенту входить в личный кабинет, в котором находится вся информация о самом клиенте, балансе карт, наличии счетов и их состоянии. Также оно предоставляет пользователю операции по переводу на другие карты и счета других банков, возможность открытия/закрытия предложений по вкладам, кредитам. Пользователи легко могут оплатить коммунальные или государственные услуги, получить выписки с счетов.

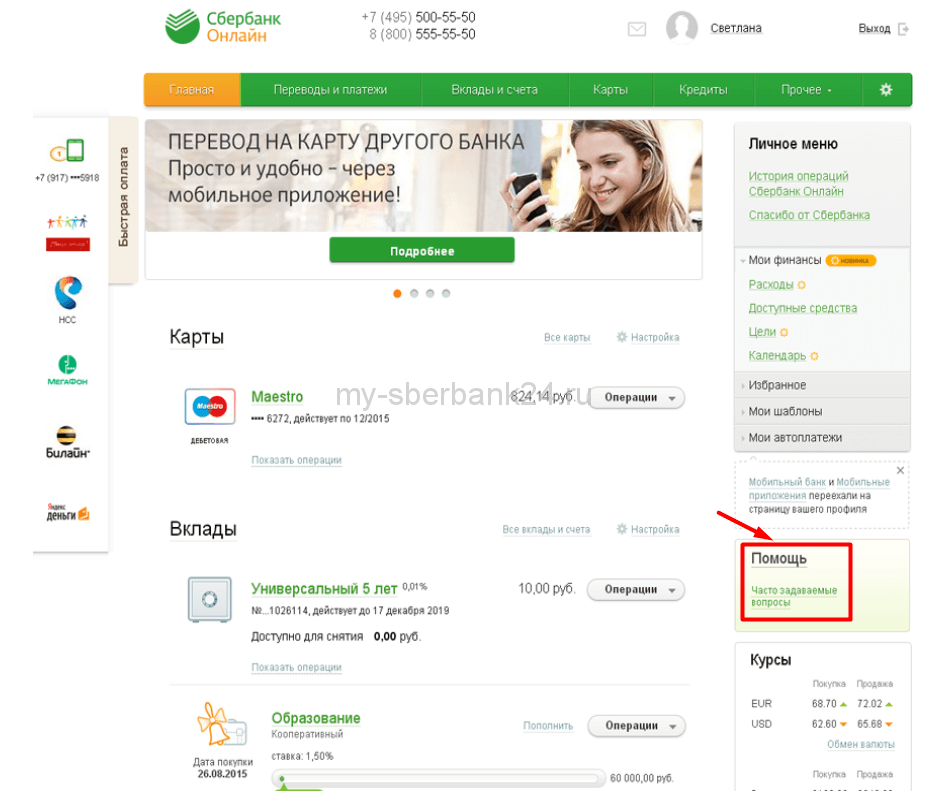


Рисунок 1 Интерфейс Личного кабинета Сбербанк

В приложении:

* Удобный и понятный интерфейс
* Возможность просмотра всех офисов, терминалов и банкоматов банка
* Широкая функциональность

**Тинькофф**

Приложение Тинькофф при входе в личный кабинет также показывает всю информацию о пользователе, его счетах, балансе карт, предоставляет все банковские операции, в том числе открытие/закрытие вкладов/кредитов. Также есть возможность просмотра и поиска среди всех произведённых финансовых операций.

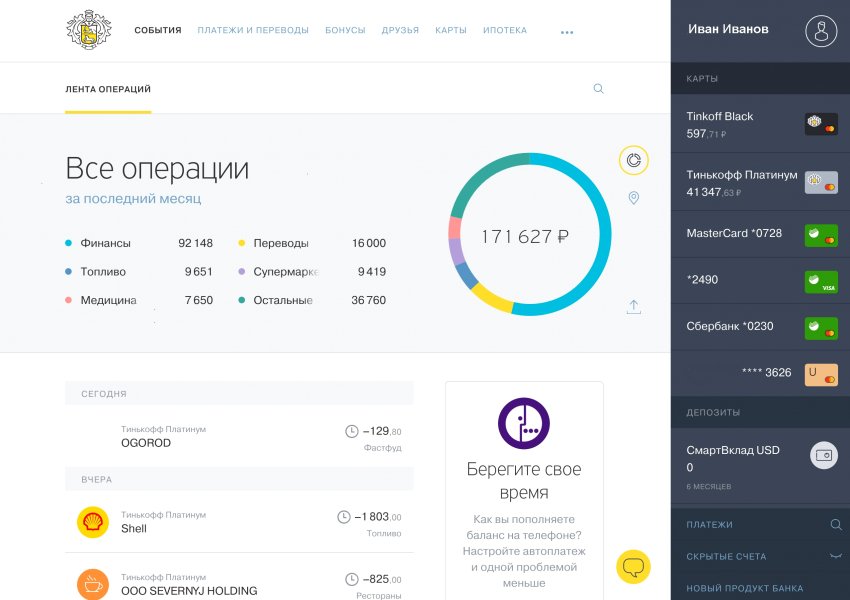


Рисунок 2 Личный кабинет Тинькофф

В приложении:

* Широкая функциональность
* Возможность заказа карты
* Удобный интерфейс

## IDEF0

Для отображения функциональности, была построена диаграмма IDEF0. Центральным бизнес-процессом является взаимодействие клиента с банковским приложением, который на схеме отображается в виде функционального блока на рисунке 3.

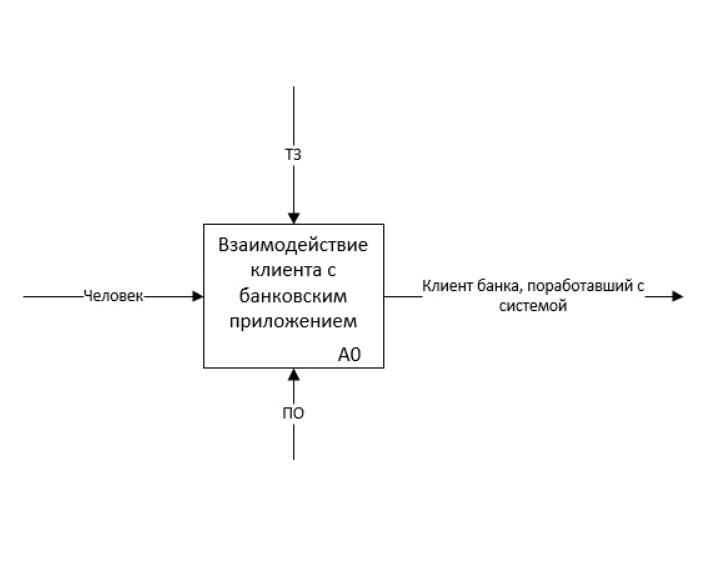


Рисунок 3 Функциональный блок A0

На рисунке 4 представлена декомпозиция функционального блока А0. Процесс взаимодействия клиента с приложение разделён на составляющие части: регистрация, авторизация, работа в приложении, выход из профиля.

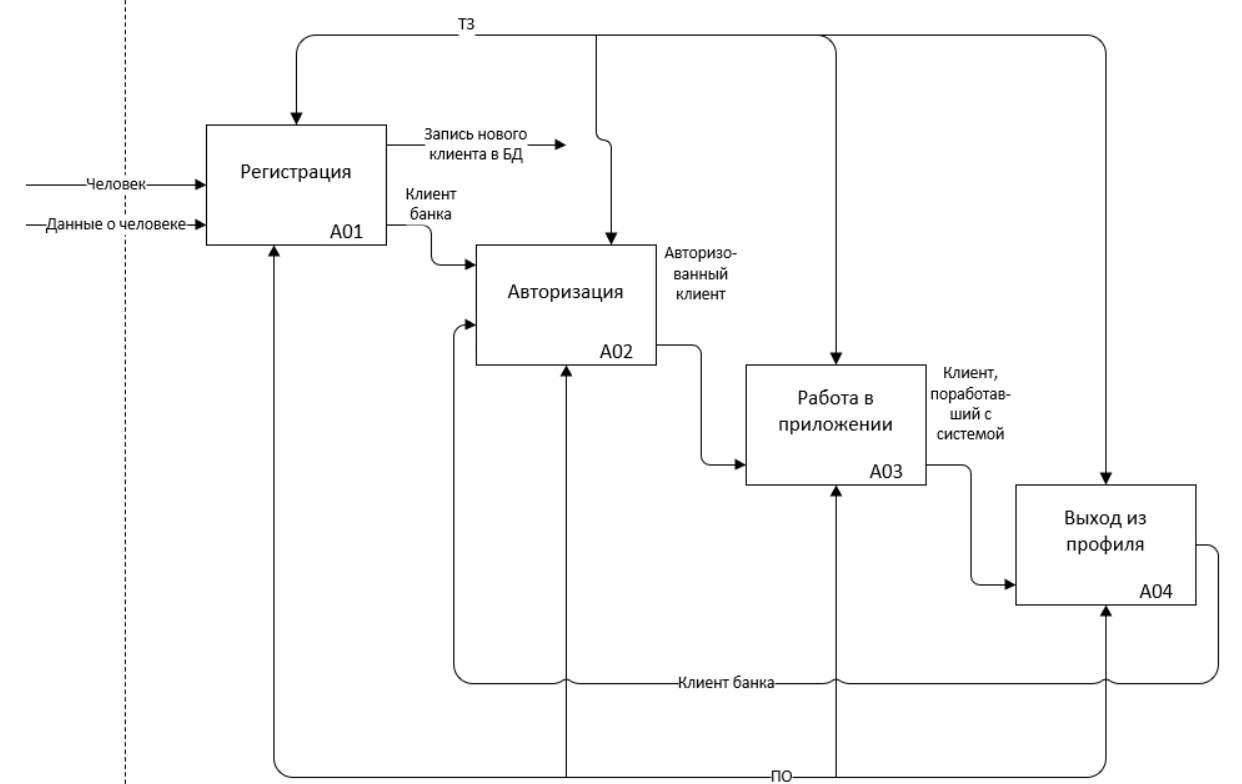


Рисунок 4 Декомпозиция блока А0

Работа в приложении подробна описана в диаграммах, которые представлены далее.

## Диаграмма прецедентов

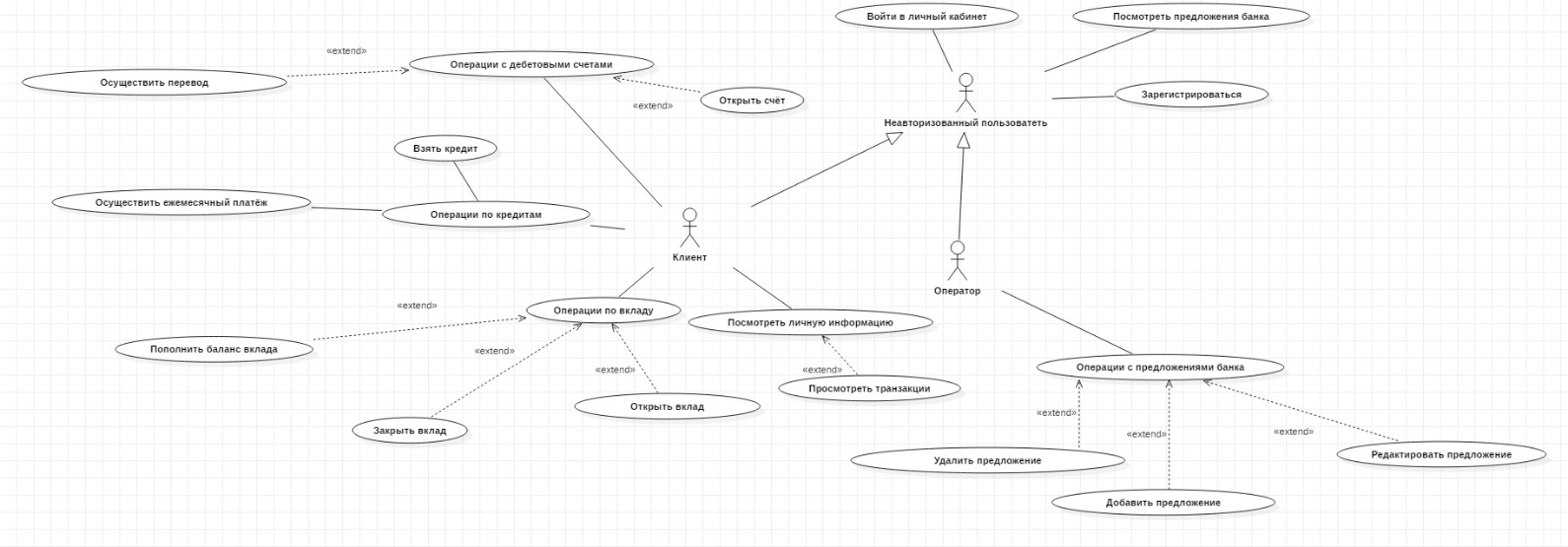
Диаграмма прецедентов показывает возможности различных пользователей в системе. Все эти варианты использования были составлены в соответствии с техническим заданием (разделы функциональных возможностей).

Рассмотри диаграмму более подробно. Изначально с веб-приложение взаимодействует неавторизованный пользователь. На главной странице он просматривает действующие приложения. Так же он может либо войти в личный кабинет, либо зарегистрироваться. С данным актёром отношением наследования связаны клиент банка и оператор. Каждая из этих сущностей имеет свой набор функций.

Клиент банка может:

* совершать операции с дебетовыми/кредитными/депозитными счетами;
* просматривать личную информацию.

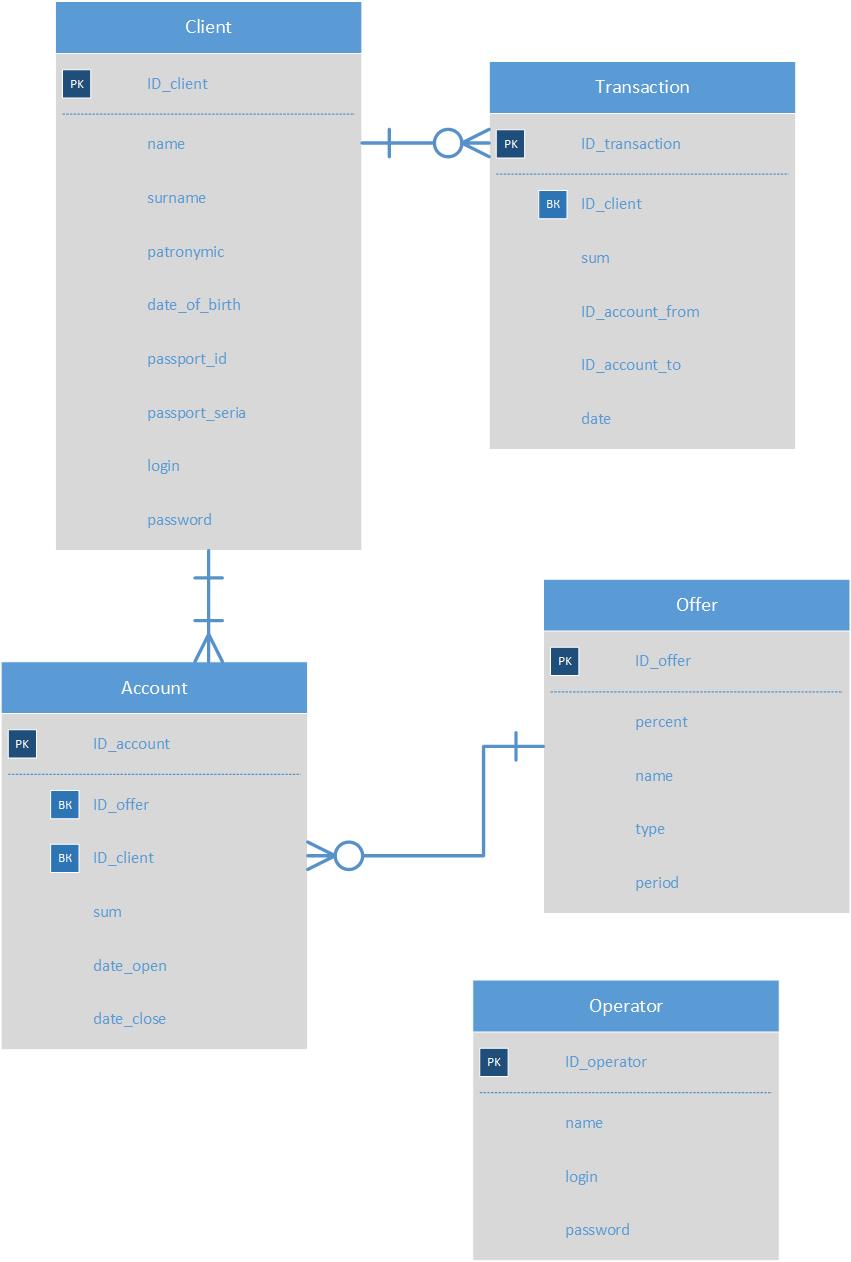
Оператор банка имеет возможность добавлять/редактировать/удалять предложения банка.



# Проектирование приложения

## ER-диаграмма

Схема БД для сущностей, которые необходимо хранить в системе. На данной диаграмме также отражены связи между таблицами.



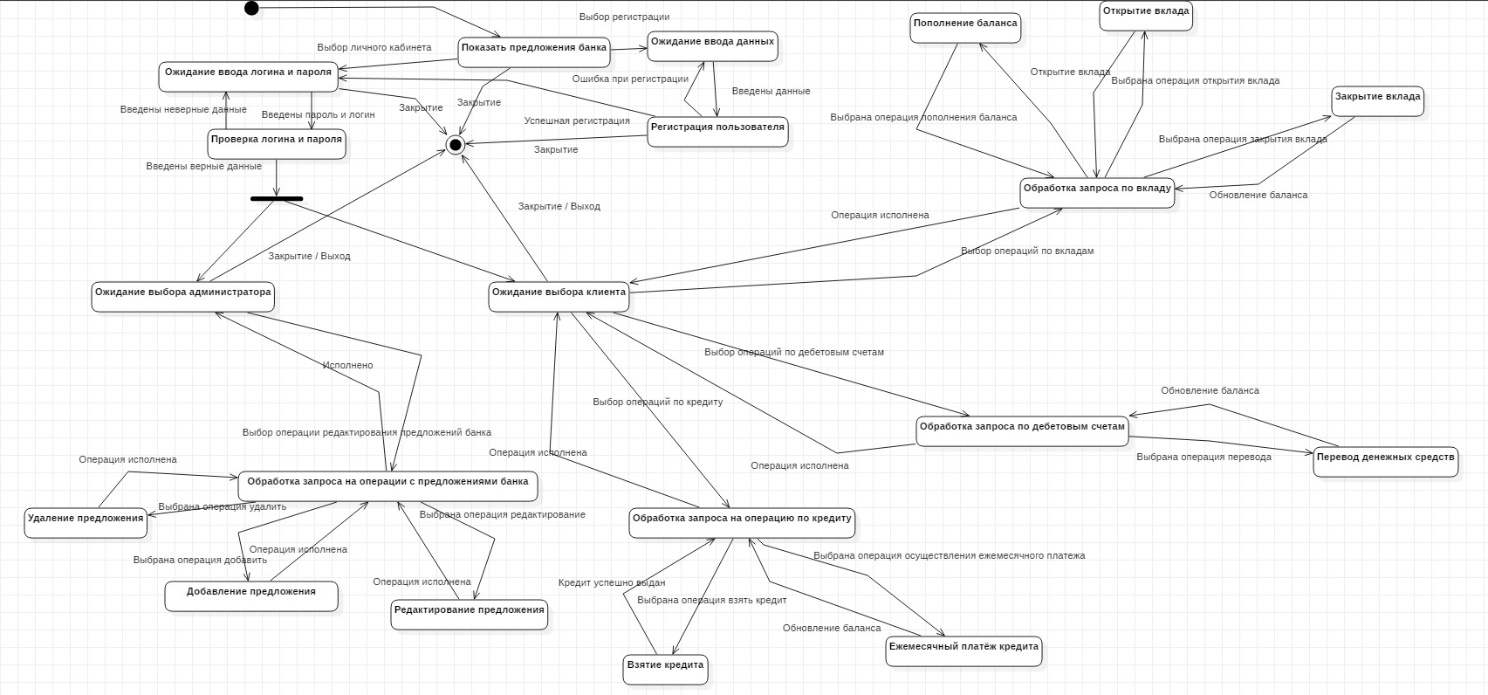
Данная диаграмма составлена для проектирования баз данных, в которых будет отсутствовать избыточность данных. В ней отражены сущности и различные отношения между ними. Опишем их.

**1. Client - Account**. Степень связи 1:m (один ко многим). Класс принадлежности сущности «Client» к сущности «Account» обязательный, и наоборот (так как изначально при регистрации заводится дебетовый счёт для клиента). Каждый клиент может иметь 1 или несколько счетов, каждый счёт может принадлежать только одному клиенту.

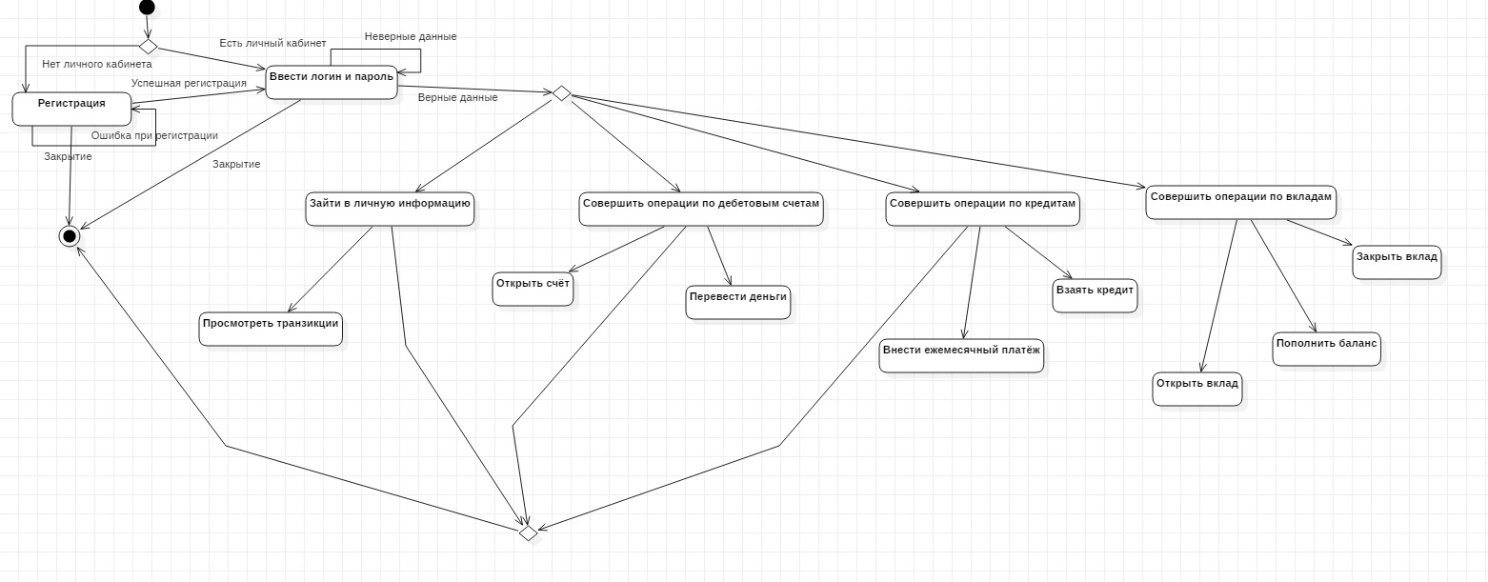
**2. Client – Transaction**. Степень связи 1:m. Класс принадлежности сущности «Client» к сущности «Transaction» необязательный, так как клиент мог не совершать переводов. А класс принадлежности сущности «Transaction» к сущности «Client» обязательный, потому что транзакция осуществляется конкретным клиентом. Каждый клиент может осуществлять большое количество транзакций, но у него может и не быть транзакций, например, на момент регистрации в системе. У каждой транзакции может быть только 1 клиент.

**3. Account – Offer.** Степень связи m:1. Класс принадлежности сущности «Account» к сущности «Offer» обязательный. А класс принадлежности сущности «Offer» к сущности «Account» не обязательный. Каждый счёт соответствует лишь одному предложению, а одному и тому же предложению могут соответствовать много разных счетов, а может и не быть ни одного соответствия.

## Диаграмма состояний

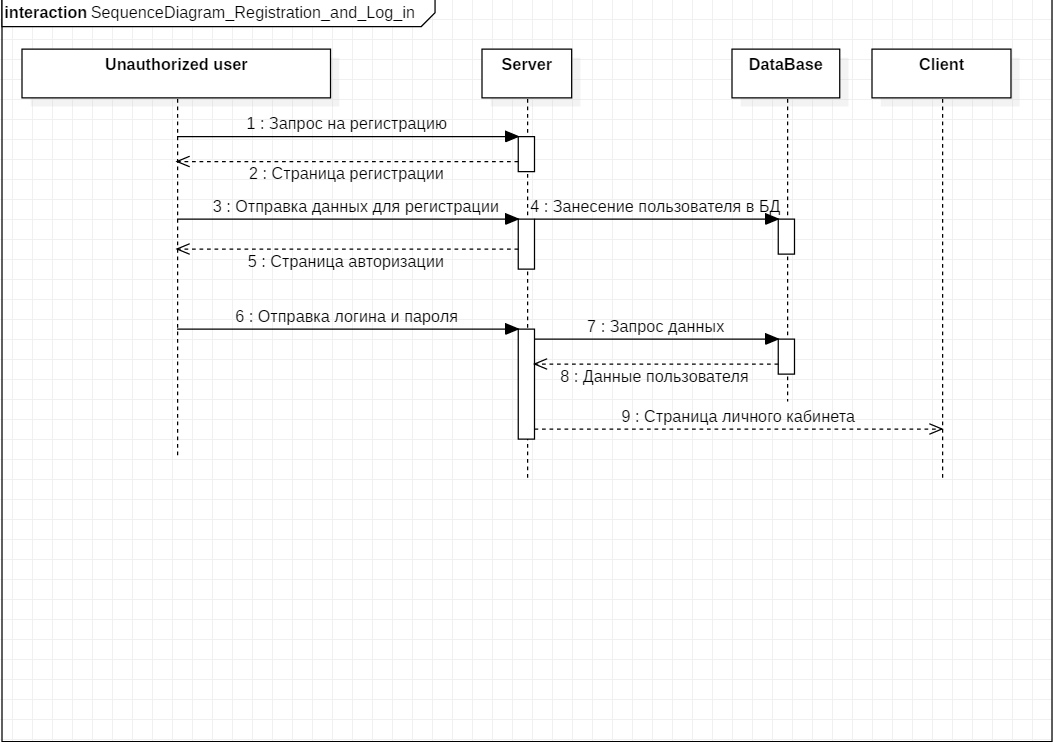


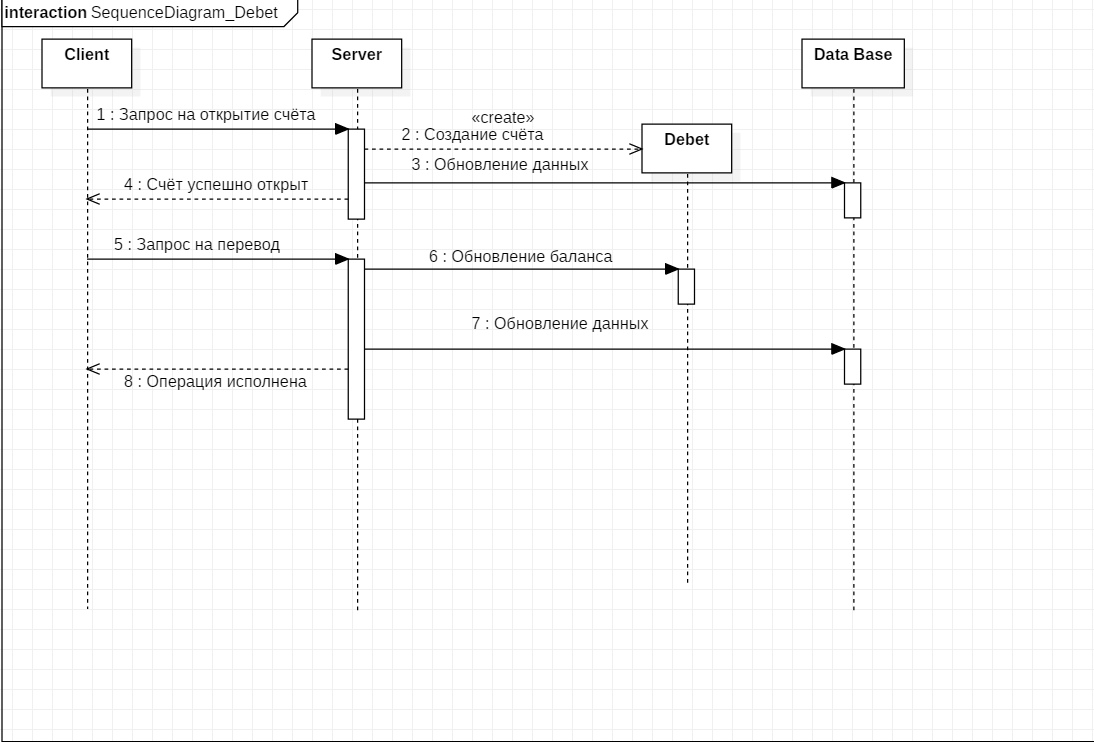
## Диаграмма активности



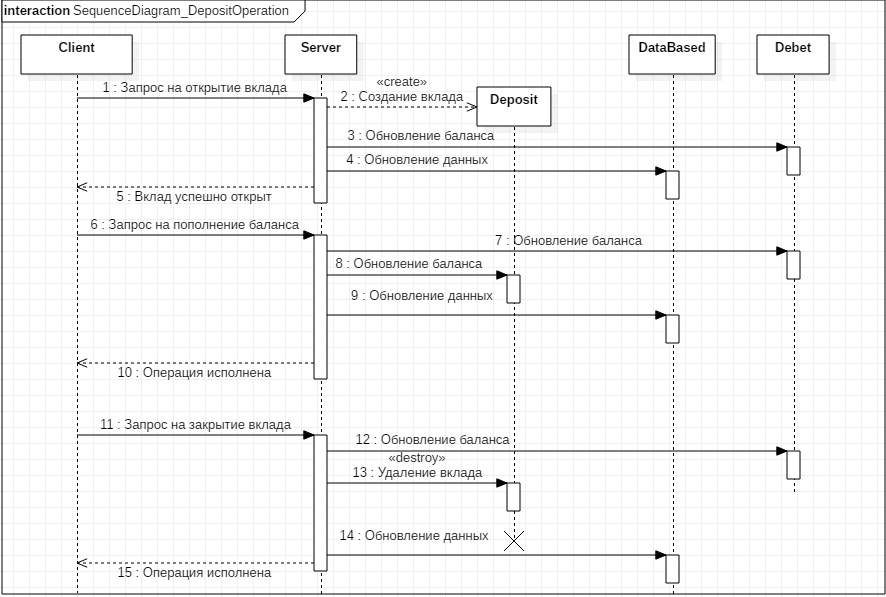
## Диаграммы последовательностей

**Диаграмма последовательности регистрации и авторизации**

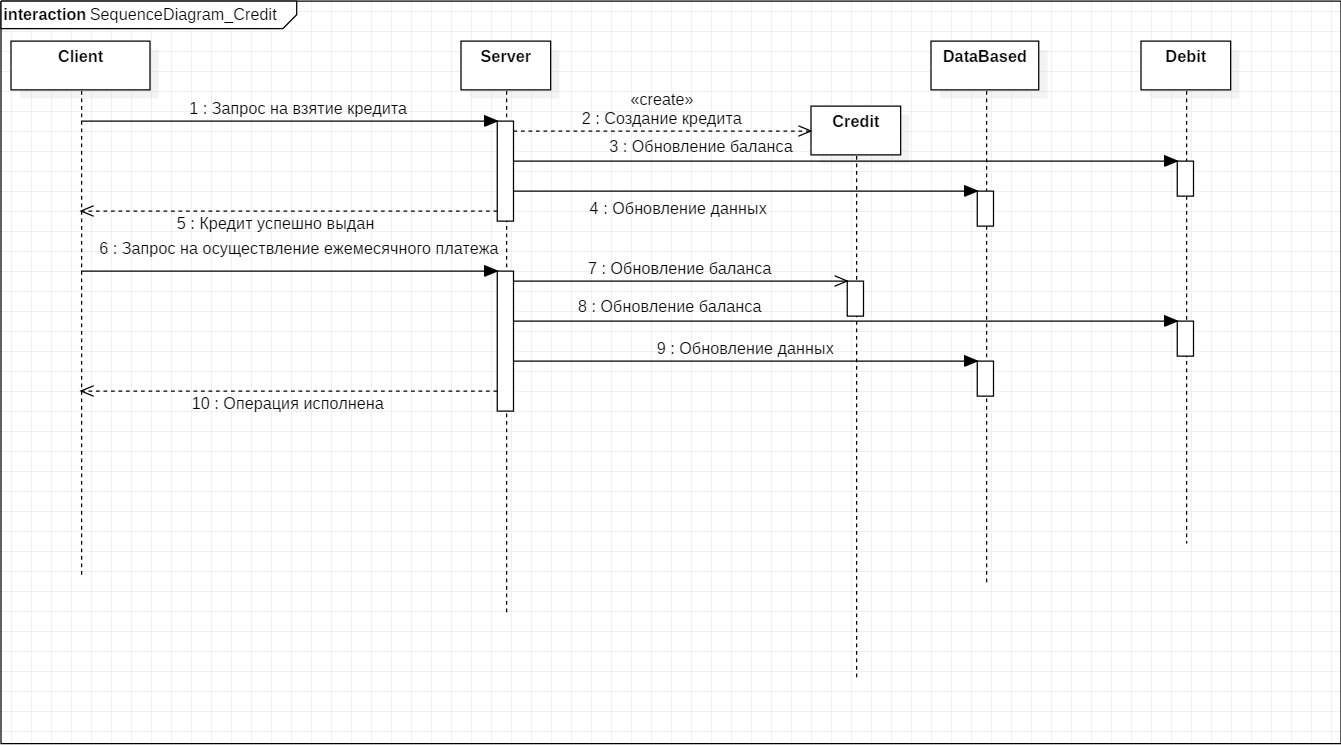


**Диаграмма последовательности операций с дебетами**

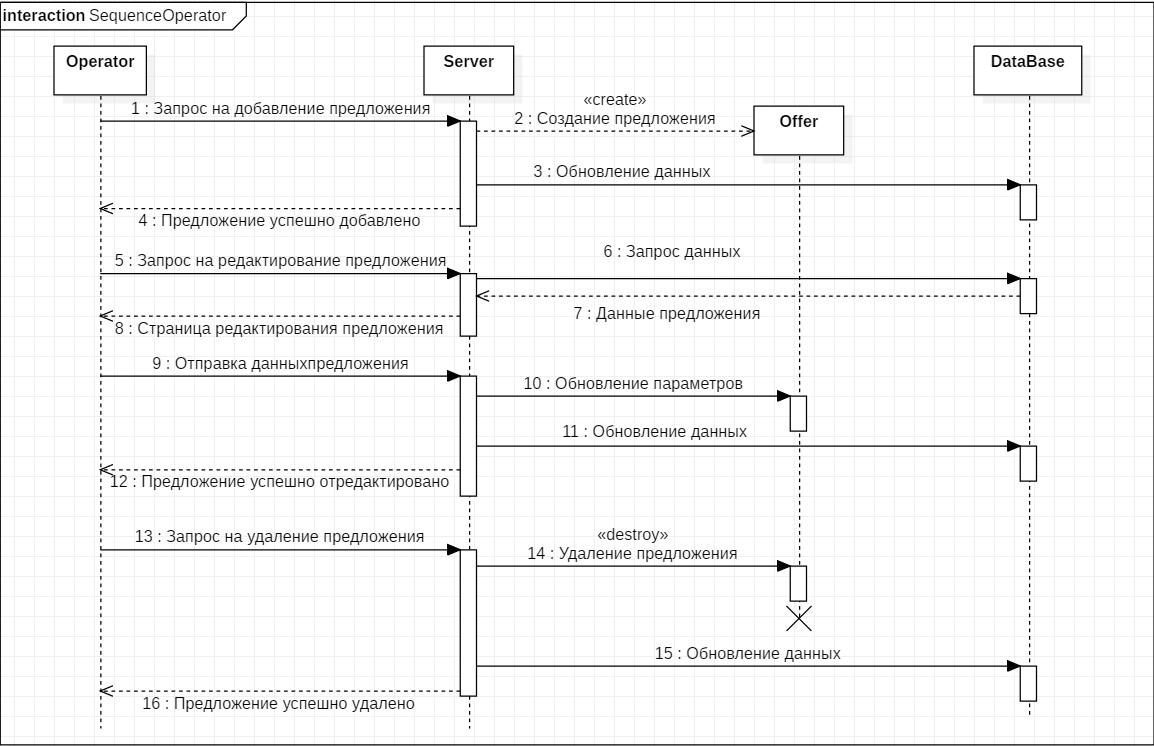
**Диаграмма последовательности операций с вкладами**



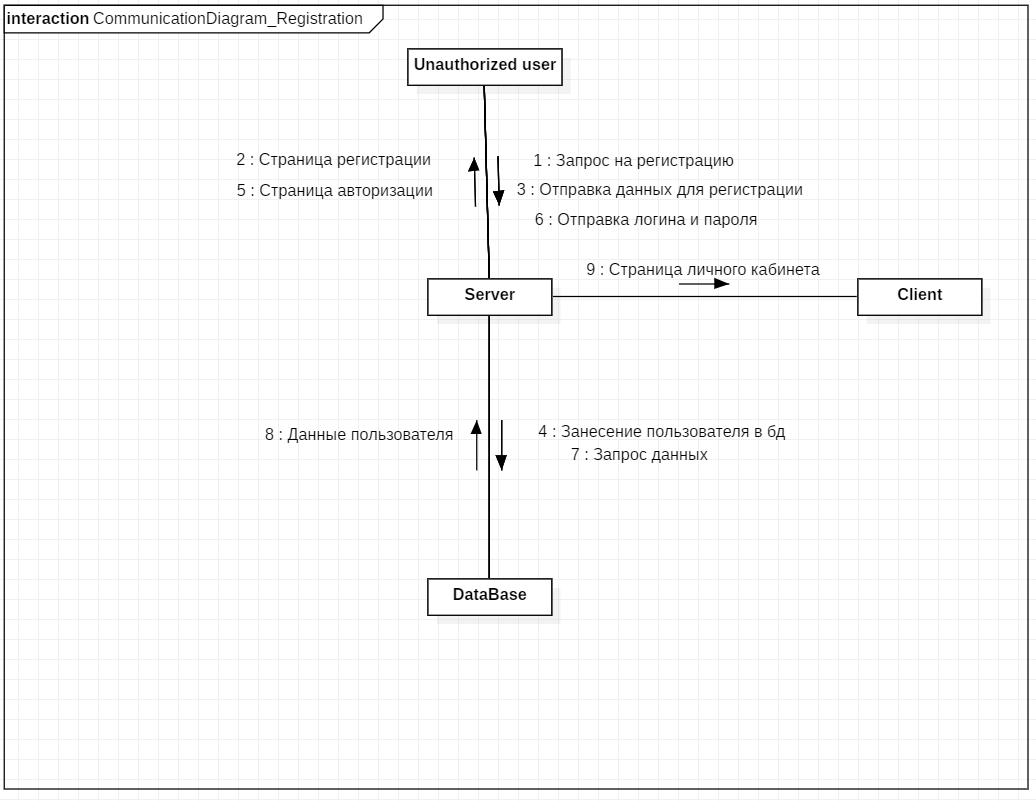
**Диаграмма последовательности с кредитами**

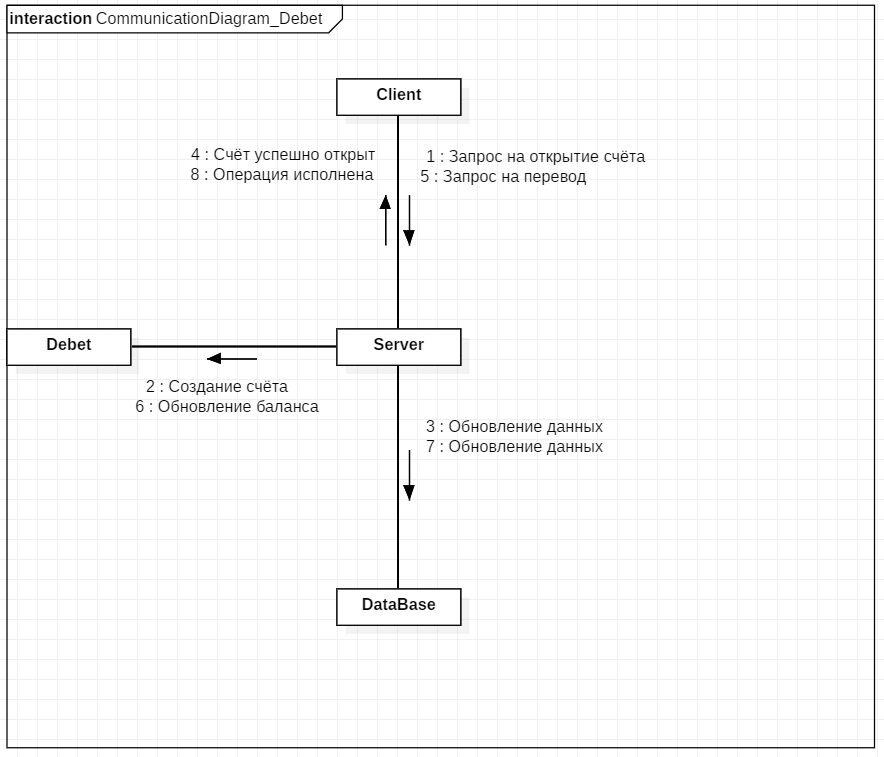


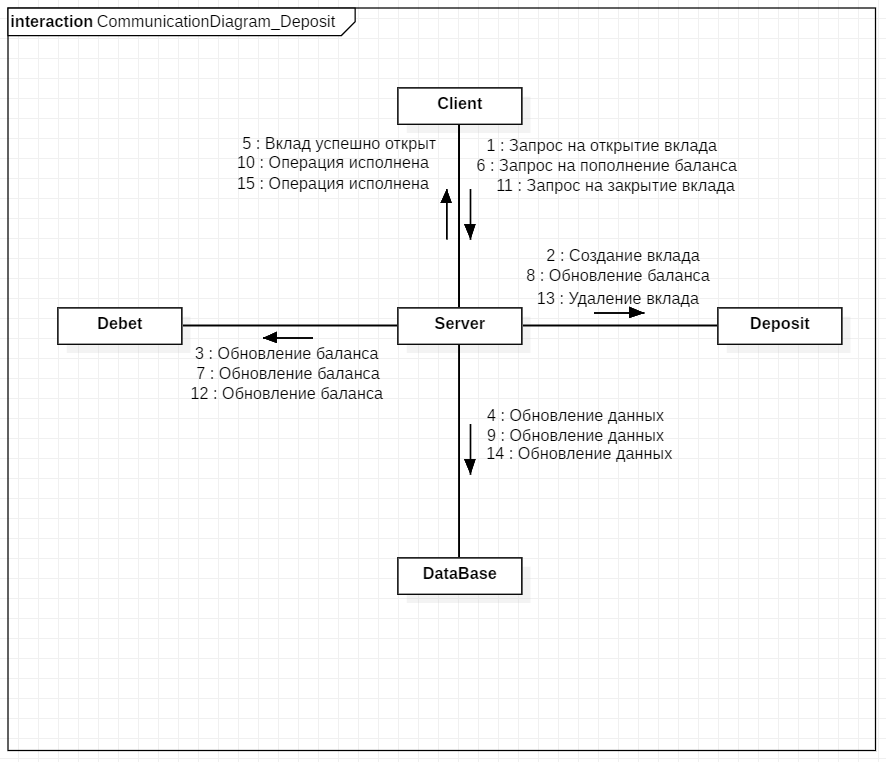
**Диаграмма последовательности оператора**

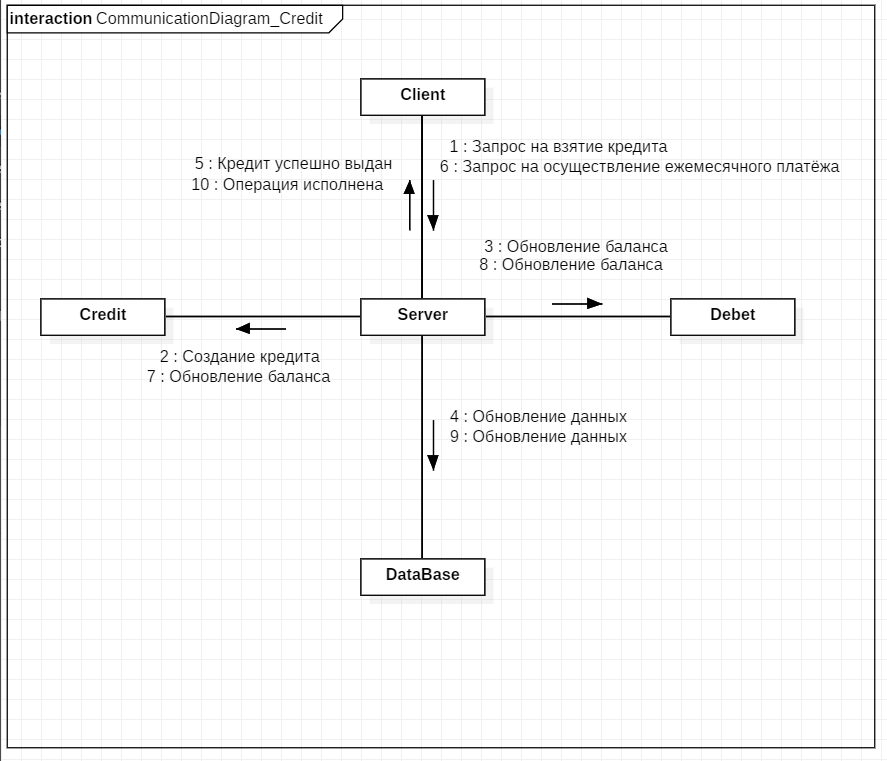


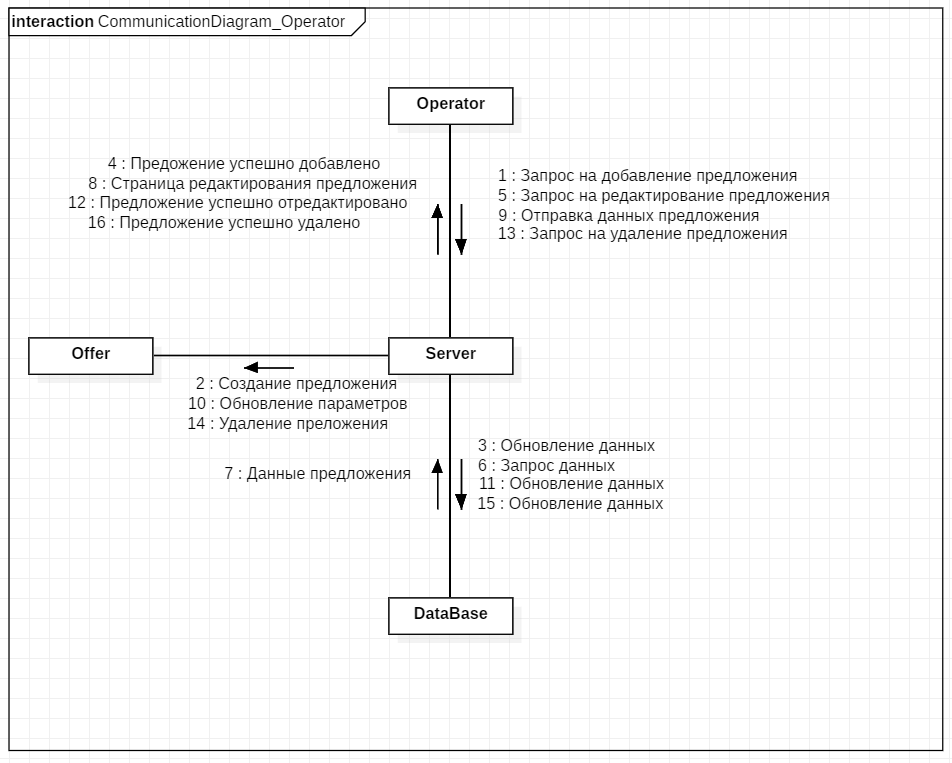
## Диаграммы коммуникации

**Диаграмма коммуникации регистрации и авторизации**

**Диаграмма коммуникации операций с дебетами**

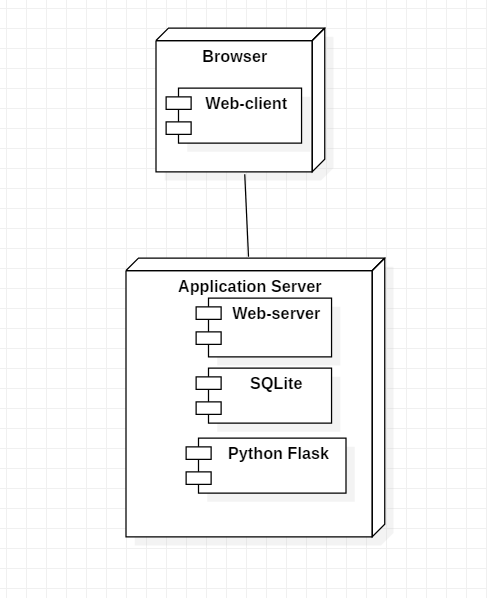
**Диаграмма коммуникации операций с вкладами**

**Диаграмма коммуникации с кредитами**

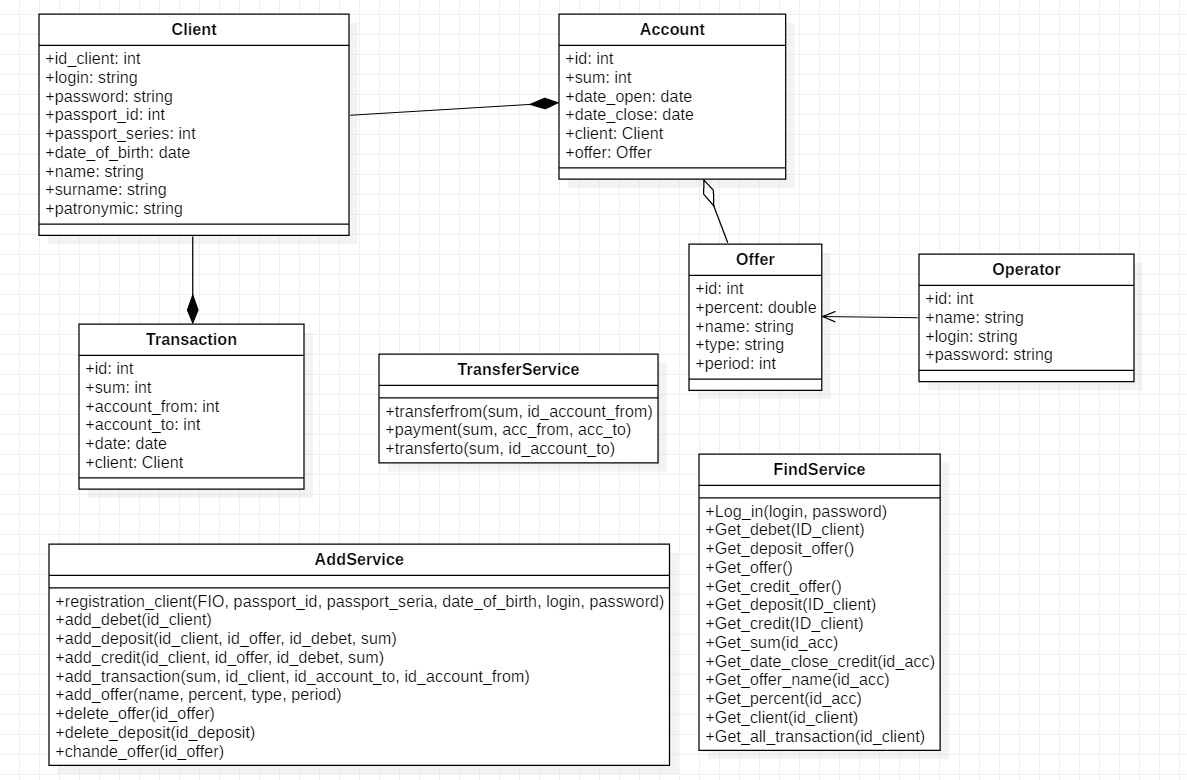
**Диаграмма коммуникации оператора**

## Диаграмма развёртывания

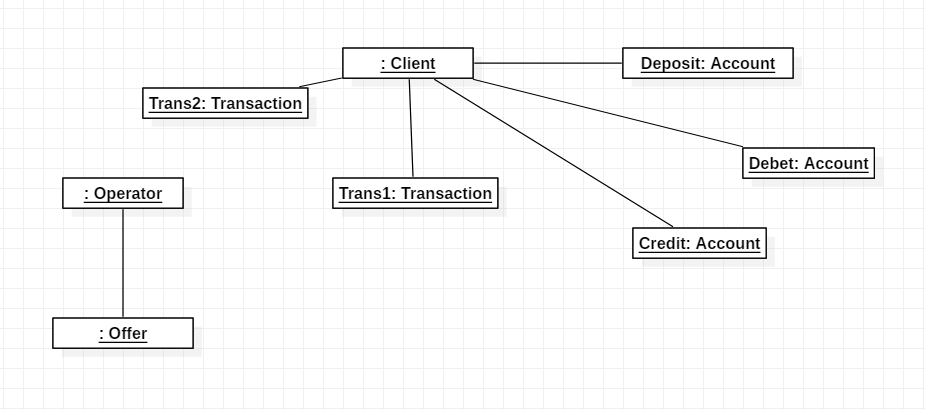
Взаимодействие между клиентом и сервером будет проводится по протоколу HTTP.



## Диаграмма классов



## Диаграмма объектов



# Реализация приложения

## Средства реализации

Для создания диаграмм и схем на этапе проектирования использовались следующие программные средства:

* StarUML
* MS Visio 2016

В качестве языка программирования для реализации приложения был выбран Python. Этот язык был выбран так как он:

* реализован практически на всех платформах
* имеет понятный и доступный синтаксис
* нетипизированный, а значит позволяет переносить программы без уточнений о размерности того или иного типа и производить различные манипуляции без особых сложностей
* имеет большое количество фреймворков, в частности для работы с веб-приложениями

Для написания программы была выбрана среда разработки PyCharm полнофункциональная IDE для Python, доступна на платформах Windows, Mac OS X и Linux, существует в бесплатном (Community) и платном (Professional) вариантах. В качестве СУБД использовалась SQLite, которая не требует настройки сервера СУБД, располагается в одном файле, что облегчает перемещение и имеет достаточно высокую скорость обработки данных.

В качестве фреймворка был выбран Flask, который имеет следующие преимущества:

* проста и гибкость в работе
* позволяет разрабатывать гибкие модули, которые потом с лёгкостью можно использовать в других проектах
* имеет возможности подключения многих библиотек

Для работ с управлением данными из реляционных БД был выбран ORM Peewee, которое автоматизирует перевод данных из реляционных БД в объекты, используемые в коде, а также позволяет писать код взаимодействия с БД на языке разработки, обеспечивает интерфейс для операций создания чтения обновления и удаления данных. Peewee ORM более лёгкая и быстрая ORM, которая может быть использована в большинстве веб-фреймворках.

В качестве архитектуры приложения была выбрана стандартная архитектура REST, которая проста и понятна в использовании. Данные, пришедшие на сервер, отправляются далее в том же виде, что и были получены, без каких-либо дополнительных прослоек. Управление информацией сервиса основывается на протоколе передачи данных. Наиболее распространенным протоколом является HTTP. Для него действие над данными задается с помощью методов: GET (получить), PUT (добавить, заменить), POST (добавить, изменить, удалить), DELETE (удалить). Таким образом, в REST сервер не запоминает состояние пользователя между запросами. И при каждом новом запросе передаётся вся информация, необходимая для выполнения той или иной операции. Веб-сервер имеет API, предоставляемый ему сервисами backend. Он использует различные функции и получает необходимые результаты в виде DTO.

# Планирование работ

## Виды работ в процессе разработки

Разработку приложения планируется в несколько этапов:

* 1. Анализ предметной области и подготовка документации приложения
  2. Проектирование приложения
  3. Разработка приложения
  4. Тестирование приложения и общее заключение

**Анализ предметной области и подготовка документации приложения**

На этапе данном этапе необходимо понять для кого реализуется приложение, какие уже существуют аналоги, какие задачи решает приложение

Для этого были написаны:

* введение
* постановка задачи
* анализ предметной области
* определение требований
* диаграмма прецедентов

**Проектирование приложения**

На этом этапе необходимо было проанализировать структуру приложения, определить её основные модули и их взаимодействие.

Для этого созданы:

* диаграмма классов
* ER-диаграмма
* диаграмма состояний
* диаграммы последовательности
* диаграммы коммуникации
* диаграмма развёртывания
* диаграмма объектов
* мокапы страниц
* функциональная схема
* ТЗ

**Разработка приложения**

На данном этапе необходимо подготовить все программные средства для создания приложения, сделать базы данных для него, реализовать все классы и сервисы.

**Тестирование приложения**

После завершения этапа разработки необходимо проанализировать решение, для этого будет проводиться тестирование, исходя из которого можно сделать общий вывод о веб-приложении.

# Заключение

# Список литературы