МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

**«РАЗРАБОТКА ПРОЕКТА ПО УПРАВЛЕНИЮ СИСТЕМОЙ ИНКАССАЦИИ»**

Студент Кислюк И. В.

Факультет Инфокоммуникационные технологии

Кафедра Программных систем

Группа К4120

Направление (специальность) 11.04.02 Программное обеспечение в инфокоммуникациях

Руководитель Иванов Сергей Евгеньевич, доцент, к. ф.-м. н.

Дисциплина Разработка и внедрение распределенных систем

Санкт-Петербург

2018 г.

Содержание

[1. Определение целей проекта и предметной области проекта 4](#_Toc509310350)

[2. Обзор информационных систем для реализации проекта 5](#_Toc509310351)

[3. Определение функциональных и нефункциональных требований 7](#_Toc509310352)

[3.1. Функциональные требования 7](#_Toc509310353)

[3.2. Нефункциональные требования 7](#_Toc509310354)

[4. Формирование образа проекта в целом 9](#_Toc509310355)

[5. Выполнение анализа бизнес-процессов. Определение сущностей и необходимых атрибутов. 10](#_Toc509310356)

[6. Определение отношений между сущностями, представление их графически. 13](#_Toc509310357)

[7. Определение атрибутов, которые будут являться уникальными идентификаторами для каждой сущности. 14](#_Toc509310358)

[8. Определение основной функциональности системы и формирование ключевых функций системы 14](#_Toc509310359)

[9. Разработка модели информационной системы в виде диаграммы потоков данных 18](#_Toc509310360)

[10. Определение структуры информационных потоков 19](#_Toc509310361)

[11. Разработка функциональной архитектуры системы 20](#_Toc509310362)

[12. Выбор архитектуры для проекта распределенной системы 21](#_Toc509310363)

[13. Разработка логической модели данных информационной системы. 23](#_Toc509310364)

[14. Выполнение нормализации логической модели данных 24](#_Toc509310365)

[15. Построение физической модели данных системы. Определение типов данных для атрибутов. Создание реляционной модели. 25](#_Toc509310366)

[16. Составление основных пунктов технического задания на проект 27](#_Toc509310367)

[1.Бизнес требования 27](#_Toc509310368)

[2.Рамки и ограничения проекта 28](#_Toc509310369)

[Заключение 31](#_Toc509310370)

# 1. Определение целей проекта и предметной области проекта

Система «Служба инкассации» – это новая система, которая заменяет текущие ручные процессы оформления заявок и составления графиков доставки и ручного распределения инкассаторов по машинам. С помощью этого приложения появиться возможность ускорения процесса регистрации клиентов в системе, упрощение процесса создания заявок. Отслеживание, учет и планирование маршрутов путем введения электронной системы упростятся, что позволит снизить нагрузку на персонал и уменьшить возможность ошибок.

# 2. Обзор информационных систем для реализации проекта

В данный момент не так много банковских организаций предоставляют открытую систему для службы инкассации, поэтому необходимость разработки открытой системы службы инкассации возрастает. Ввиду закрытости существующих информационных систем очень сложно проводить анализ и выявление достоинств и недостатков текущих решений. Среди рассмотренных банковских организаций были выбраны «Сбербанк», «Альфа», «Банк СПб». У двух последних система службы инкассации представлена телефонным справочником и необходимо осуществлять заявку вручную. Пример приведен на рисунке 1.

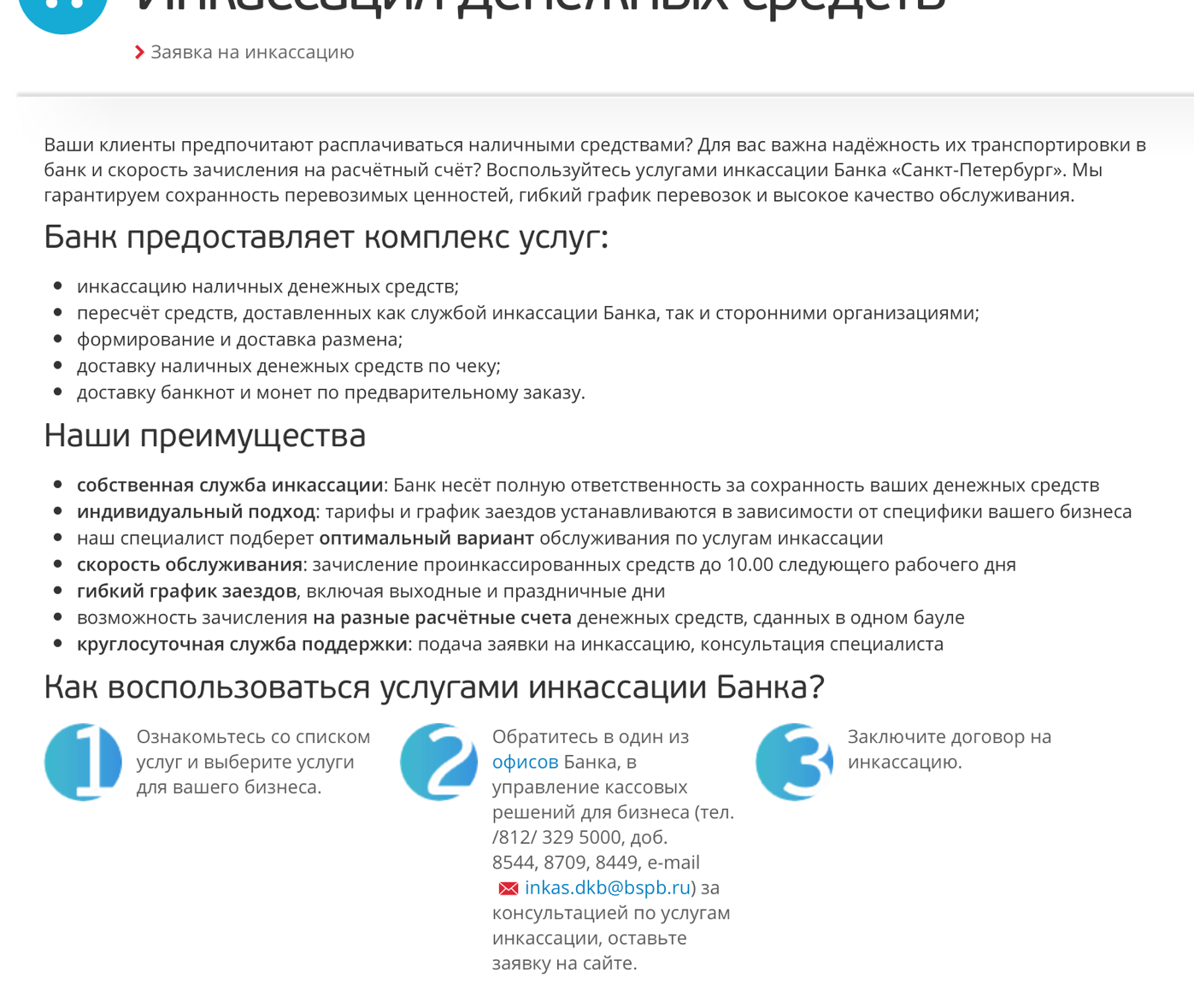


Рисунок 1 – пример службы «Банка СПб»

В случае «Сбербанка» на сайте сразу предоставлена система для ввода необходимых данных для запроса инкассации. Пример приведен на рисунке 2.

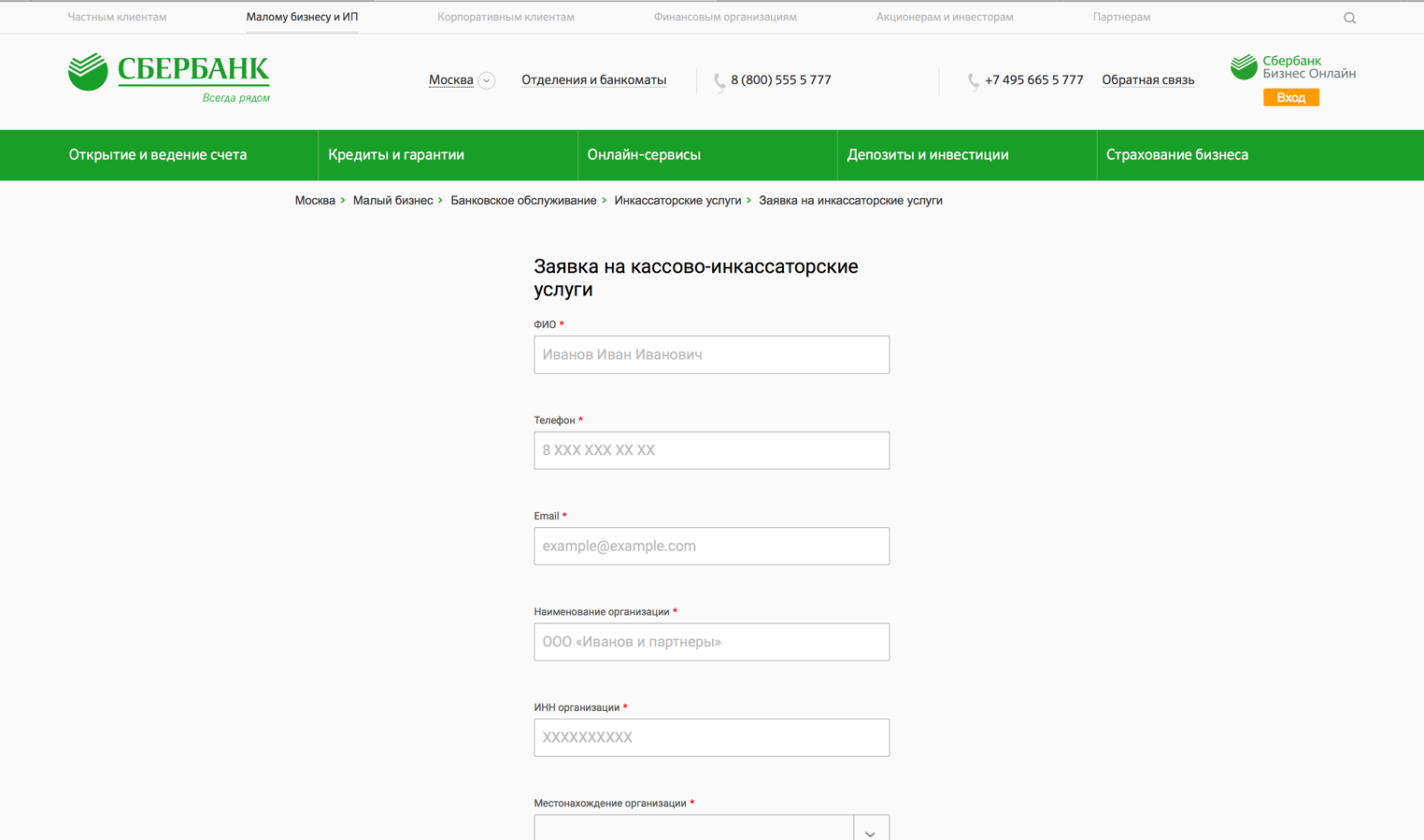


Рисунок 2 – Пример службы от «Сбербанка»

# 3. Определение функциональных и нефункциональных требований

## 3.1. Функциональные требования

Для полноценной работы системы требуется реализовать следующие функциональные требования:

* Система должна предоставлять возможность создания заявки на инкассацию со стороны клиента
* Система должна позволять отслеживать состояние заявки
* Система должна предоставлять возможность регистрации клиентов в системе
* Должна быть возможность добавления объектов со стороны как клиентов, так и сотрудников
* Система должна предоставлять возможность создания и отправки отзывов любым участником процесса
* Система должна позволять пользователям генерировать и отправлять отчеты в удобном им формате

## 3.2. Нефункциональные требования

В информационной системе нефункциональные требования обычно представлены внешними интерфейсами, атрибутами качества и ограничениями системы. Для наиболее эффективной работы информационной системы следует выделить следующие атрибуты качества:

* *Требования к производительности.* Контент, генерируемый системой, должен полностью загружаться за фиксированное время при соблюдении скоростных условий сети доступа. Система должна выводить пользователю сообщение о подтверждении деятельности после того, как пользователь отослал информацию системе.
* *Требования к безопасности.* Обеспечение защиты системы от базовых атак на веб-приложения, использование шифрования по протоколу https. Использование anti-forgery токенов при отправке контента.
* *Требования к доступности.* Система должна быть в постоянном доступе за исключением времени планового обслуживания. Сбои в системе должны логироваться для дальнейшего исследования и устранения.
* *Требования к надежности.* При преждевременном разрыве соединения между пользователем и системой уже внесенная информация не должна нарушать целостность транзакции при повторном подключении. Система должна иметь постоянно обновляющуюся резервную копию БД. Необходимо предусмотреть возможность подключения дополнительных средств развёртывания для сглаживания нагрузки.
* *Требования к визуальному оформлению.* Ограничения дизайна интерфейсов и реализациисистемы должны быть заранее обговорены с заказчиком. Интерфейс должен быть удобен пользователю в любое время: отсутствие необоснованно ярких картинок, резких и интуитивно непонятных переходов и загруженных страниц.
* *Требования к документации.* Документация информационной системы по дизайну, коду и сопровождению должна соответствовать установленным стандартам и пожеланиям заказчика.

# 4. Формирование образа проекта в целом

На основе имеющихся сведений о предметной области проекта и требований к нему можно определить задачи, выполнение которых приведёт к успешному завершению проекта. Информационная система службы инкассации представляет собой распределённую систему обработки пользовательского взаимодействия, позволяющую автоматизировать бизнес-процессы и получить из этого выгоду для всех пользователей системы. Осуществление проекта заключается в реализации нового функционала согласно пожеланиям заказчика. Для этого требуется провести анализ предметной области, направленный на выявление взаимодействующих сущностей и процессов между ними, которые будут соответствовать определённым действиям со стороны всех субъектов системы. Далее следует выбрать оптимальные средства ведения проекта, разработки и тестирования функционала, а также провести начальное внедрение и обеспечение сопровождения информационной системы. После выполнения этих пунктов проект будет считаться завершённым.

# 5. Выполнение анализа бизнес-процессов. Определение сущностей и необходимых атрибутов.

Анализ бизнес-процессов позволяет смоделировать базовое поведение информационной системы, выделяя основные сущности системы (как связь взаимодействующих субъектов), действия, выполняемые этими сущностями, и влияние между ними.

В процесс изучения предметной области выбранной тематики работы, были выявлены следующие существительные: банковская система, банк-эмитент, клиент в виде юридического лица, служба инкассации, дневная выручка, купюры, сумма размена и им подобные.

Для построения логической модели необязательно использовать абсолютно все существительные. Определим сущность для каждого существительного, который будет участвовать в нашей логической модели.

Первой сущностью для системы инкассации будет административный объект, в котором будет проводиться инкассация. Назовем этот объект – Object. Объект будет обладать следующими характеристиками, такими как, описание, адрес, тип данного объекта (например, пользовательский или банкомат), уникальным кодом и прочими характеристиками. Определим следующие сущности.

- Сущность заявки на плановую инкассацию – Order. В бизнес-процессе пользователи могут создавать заявки на инкассацию, в которых указывают необходимый адрес и время, количество размена или доставку каких-либо ценных ресурсов. Например, доставка золота в магазин ювелирных украшений.

- Сущность пользователя для инкассации – User. Представляет наиболее важную часть бизнес-процесса. Пользователями являются абсолютно все участники процесса: непосредственно пользователи, менеджеры, инкассаторы.

- Сущность отзыва о приложении – Appeal. После проведения инкассации или в случае поломки, любой пользователь может оставить жалобу или предложение, а также оценить работу системы.

- Сущность адреса – Address. Представление адреса, которое содержит регион, город, улицу, и прочие необходимые атрибуты для адреса.

Остальные сущности будут представлены в логической модели в DataModeler.

Принцип бизнес-процесса следующий – пользователь регистрируется в приложении, после его до и пытается оформить заявку на инкассацию в свою фирму. Предположим, что этой фирмой будет магазин. Далее ему необходимо иметь номер договора с банком, который будет проводить инкассацию. Вместе с договором банка пользователь получает номер своего объекта. После этого пользователь, авторизовавшись в приложении может добавить свой объект в список объектов и подать заявку на плановую инкассацию этого объекта. Также может подать заявку на временный отказ от плановой инкассации. После этого пользователь может оставить оценку о работе приложения. Схематичное описание сущностей представлено в таблице 1.

Таблица 1 – Сущности, их атрибуты и описание

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Описание |
| Object | - code  - state  - object\_type  - description  - address\_id  - pass | Сущность объекта, в котором будет проводиться инкассация |
| Order | - id  - object\_code  - name  - phone\_number  - date  - description  - user\_id  - delivery\_attributes\_id | Сущность заказа. Содержит себе важную информацию о времени и месте, также о необходимом количестве размена, если таковое потребуется |
| User | - id  - name  - surname  - passport  … | Сущность пользователя, которая описывает его тип по отношению к системе и прочие характеристики, необходимые для работы системы инкассации |
| Appeal | - id  - user\_id  - type  - description  - rating  - created\_date  - object\_code | Сущность обращения пользователя в жалобой или отзывом в систему |
| Address | - id  - postcode  - city  … | Сущность адреса, которая предоставляет всю информацию по заданному объекту |

# 6. Определение отношений между сущностями, представление их графически.

С целью определения отношений между сущностями воспользуемся матрицей отношений. В таблице 2 представлена матрица отношений между сущностями.

Таблица 2 – матрица отношений между сущностями

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Объект** | **Пользователь** | **Заявка** | **Обращение** | **Адрес** |
| **Объект** |  | **+** | **+** |  | **+** |
| **Пользователь** | **+** |  | **+** | **+** |  |
| **Заявка** | **+** | **+** |  |  | **+** |
| **Обращение** |  | **+** |  |  |  |
| **Адрес** | **+** |  | **+** |  |  |

Данная модель представляет собой сущности, наделённые атрибутами, и взаимосвязанные между собой различными типами связей, которые определяют взаимоотношения различных сущностей.

# 7. Определение атрибутов, которые будут являться уникальными идентификаторами для каждой сущности.

Для каждой сущности определены стандартные уникальные идентификаторы в качестве дополнительного атрибута, значение которого позволит однозначно определить ту или иную реализацию типа сущности. Также наличие уникального идентификатора упростит создание моделей в дальнейшем.

# 8. Определение основной функциональности системы и формирование ключевых функций системы

Для определения основной функциональности системы и формирование ключевых функций системы распишем ключевые функции, определенные пунктом выше, через основную составляющую функциональность.

Таблица 3 – Функционал «Оформление заявки»

|  |  |
| --- | --- |
| **Оформление заявки** | **Описание** |
| Регистрация | Клиент должен иметь возможность зарегистрировать. |
| Добавление желаемого объекта | Клиент должен иметь возможность добавить существующий объект в систему |
| Составление заявки | Должно быть правильно указано необходимое место и время, доступное для доставки размена |
| Отправка заявки | Должна быть проверка заявки на максимальное количество и правильно указанные контактные данные |

Таблица 4 – Функционал «Регистрация объекта»

|  |  |
| --- | --- |
| **Регистрация объекта** | **Описание** |
| Пароль | Система должна уметь проверять совпадение пароля |
| Адрес | Система должна проверять наличие адреса и его корректность |
| Пользователь | Система должна иметь возможность проверить что пользователь, регистрирующий объект является сотрудником |

Таблица 5 – Функционал «Составление отзыва»

|  |  |
| --- | --- |
| **Составление отзыва** | **Описание** |
| Пользователь | Система должна уметь определять пользователя |
| Оценка | Система должна давать выбор пользователю ставить или не ставить оценку |
| Тип отзыва | Система должна давать выбор пользователю между отзывом или жалобой |

Таблица 6 – Функционал «Просмотр отзывов пользователей»

|  |  |
| --- | --- |
| **Просмотр отзывов пользователей** |  |
| Отзывы | Сотрудник должен иметь возможность получить последние отзывы для составления отчета по работе системы |

# 9. Разработка модели информационной системы в виде диаграммы потоков данных

Модель потоков данных основа на инфологической модели и реализуется при помощи диаграммы потоков данных. На диаграмме, представленной на рисунке 3, указаны основные действующие лица системы, порождающие и принимающие потоки данных.

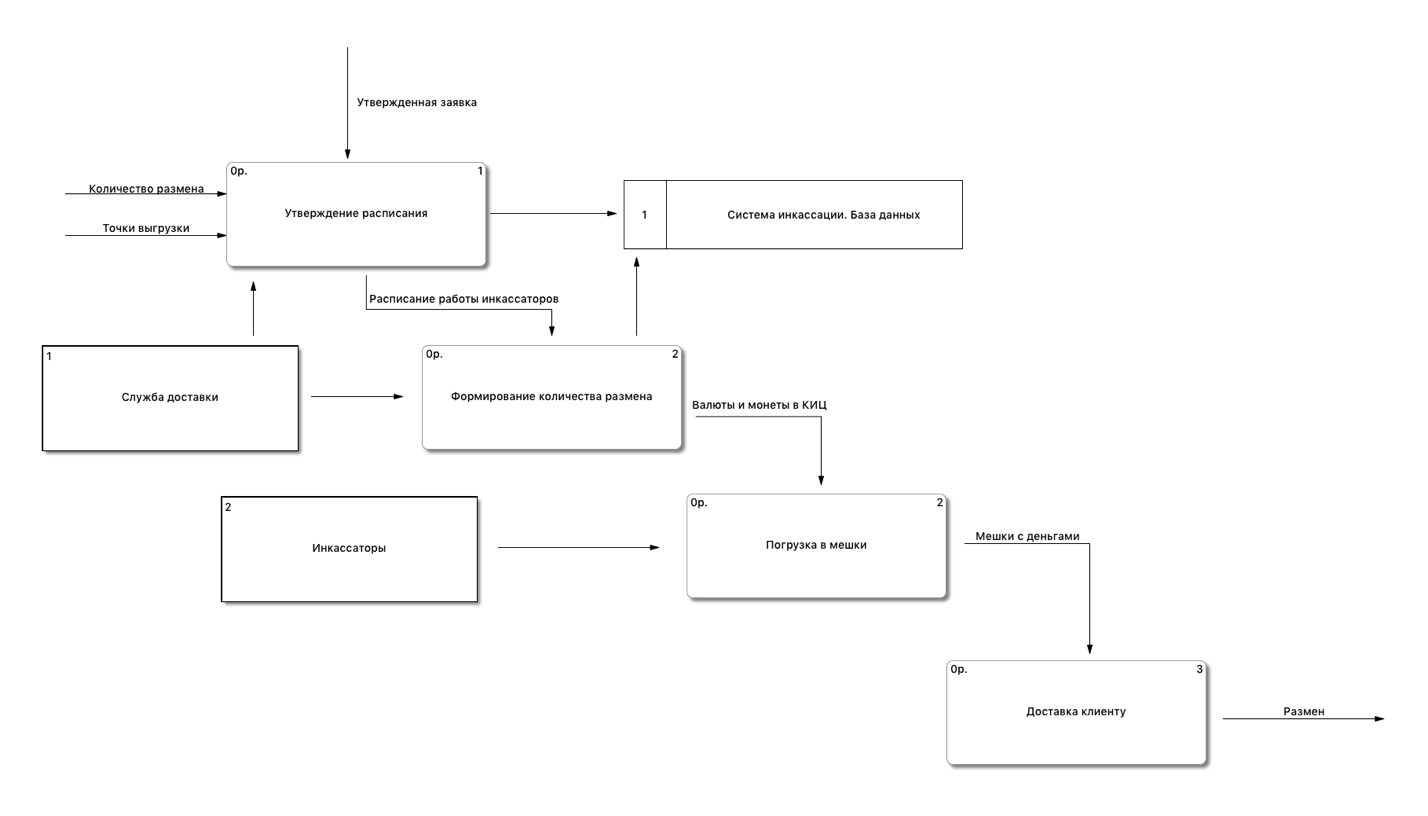
****

Рисунок 3 – Диаграмма потоков данных

# 10. Определение структуры информационных потоков

Структура информационных потоков в информационной системе службы инкассации состоит из следующих элементов, представленных в таблице 7.

Таблица 7 – Структура информационных потоков

|  |  |  |
| --- | --- | --- |
| **Название элемента** | **Описание** | **Объекты системы** |
| Издатель информации | Элемент системы, порождающий потоки информации | Пользователь (клиент, инкассатор) |
| Обработчик информации | Элемент системы, получающий промежуточную информацию, для её обработки и дальнейшей передачи | Клиентское приложение, сервер |
| Получатель информации | Элемент системы, получающий конечную информацию (не подлежащую изменению или удалению) | Пользователь (пользователь, менеджер, администратор, инкассатор) |
| Хранитель информации | Элемент системы, хранящий конечные состояния потоков информации в любом виде для последующего использования | База данных |

# 11. Разработка функциональной архитектуры системы

Разработка функциональной архитектуры данной системы может быть представлена графически. Для этого определения выступает инфологическая модель данных, представленная на рисунке 4.

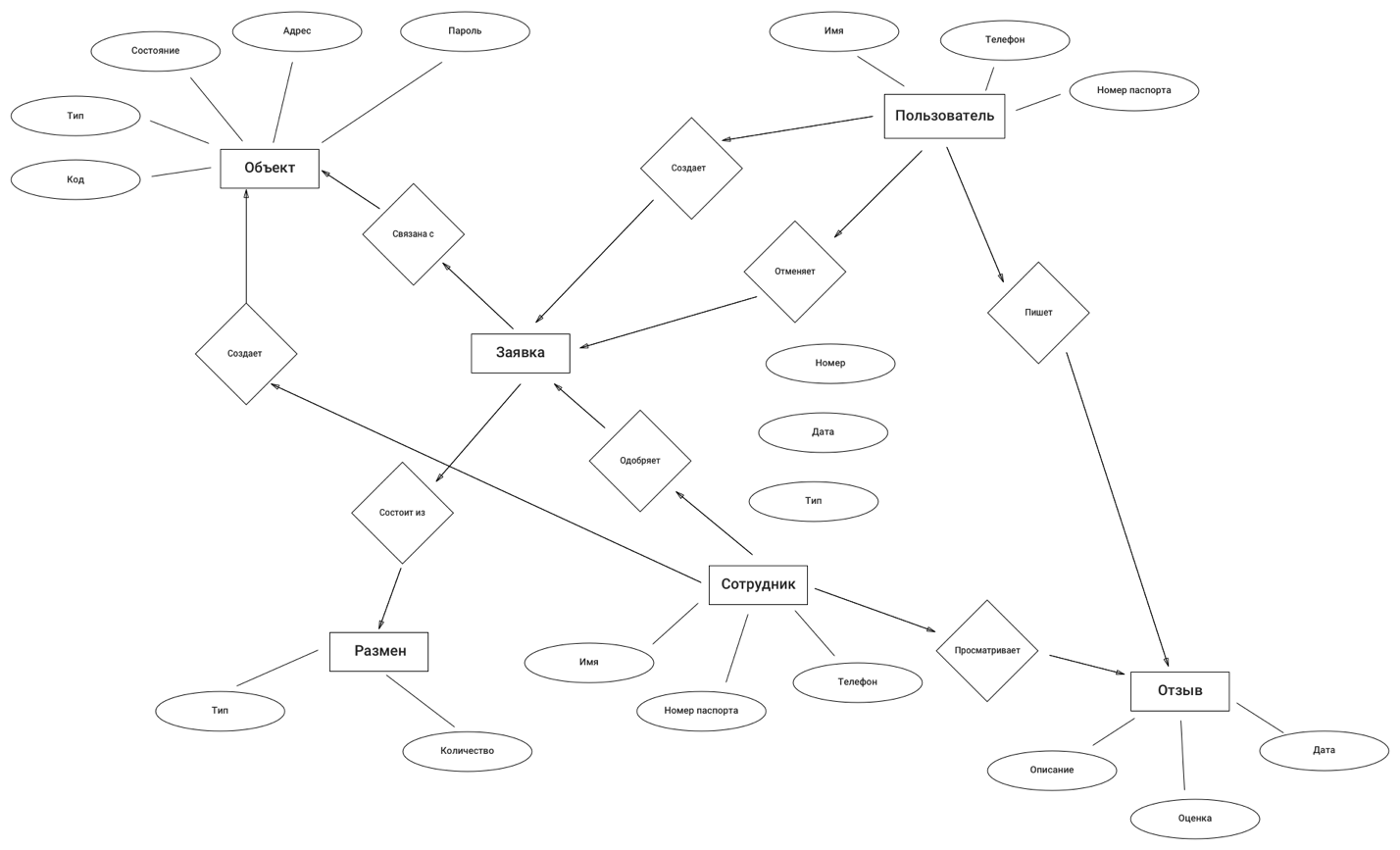


Рисунок 4 – Инфологическая модель

# 12. Выбор архитектуры для проекта распределенной системы

Для реализации данного проекта понадобится реализация серверной и клиентской частей. Для клиентской части будут использоваться соответствующие технологии:

* HTML – язык разметки документов во всемирной паутине;
* CSS – язык таблиц каскадных стилей;
* JavaScript – язык для управление контентом на стороне пользователя.

Для реализации серверной части будет использоваться серверный JavaScript – технология *Node.js*. Это программная платформа, которая преобразует JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript общаться с устройствами ввода / вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках, предоставляя им вызовы из кода JavaScript. В основе лежит событийно-ориентированное и асинхронное программирование с неблокирующим вводом / выводом. Исходя из того, что на данный момент технология активно развивается и поддерживается постоянно-растущим сообществом, именно она будет выбрана для реализации серверной части. Также технология Node.js прекрасно работает в связке с базой данных, которая поддерживает реляционность, а также хранение произвольных данных – *MongoDB.* Выбранная база данных предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядро является очень гибким и работает при подключении базы данных к приложениям через драйверы MongoDB. Существует широкий выбор доступных драйверов, поэтому легко найти драйвер, который будет работать с требуемым языком программирования. Поскольку изначально система MongoDB не была разработана для обработки моделей реляционных данных, могут возникнуть проблемы производительности, если использовать её таким образом. Однако, движок предназначен для обработки различных данных, которые нельзя отнести к реляционным, и может хорошо справляться там, где другие движки работают медленно или бессильны.

*Достоинства:*

* скорость и простота в использовании;
* поддержка json и традиционных документов NoSQL;
* данные любой структуры могут быть сохранены/прочитаны быстро.

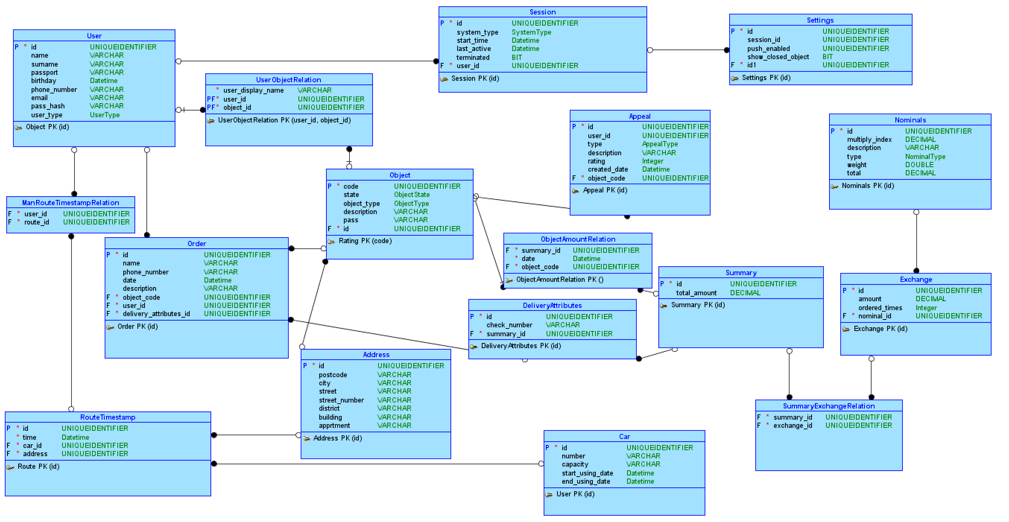
*Недостатки:*

* SQL не используется в качестве языка запросов;
* сложность установки.

Определив технологии разработки необходимо приступать к разработке логической модели данных.

# 13. Разработка логической модели данных информационной системы.

Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения. На рисунке 5 показана возможная логическая модель данных информационной системы службы инкассации.

Рисунок 5 – Логическая модель данных

# 14. Выполнение нормализации логической модели данных

В процессе выполнения работы была разрешена связь «многие ко многим» при помощи суррогатной таблицы. Таким образом, была проведена нормализация логической модели данных. Также была затронута сущность маршрута, которая раньше содержала две связи на таблицу адресов, в процессе нормализация данная связь была устранена. Пример показан на рисунке 6.

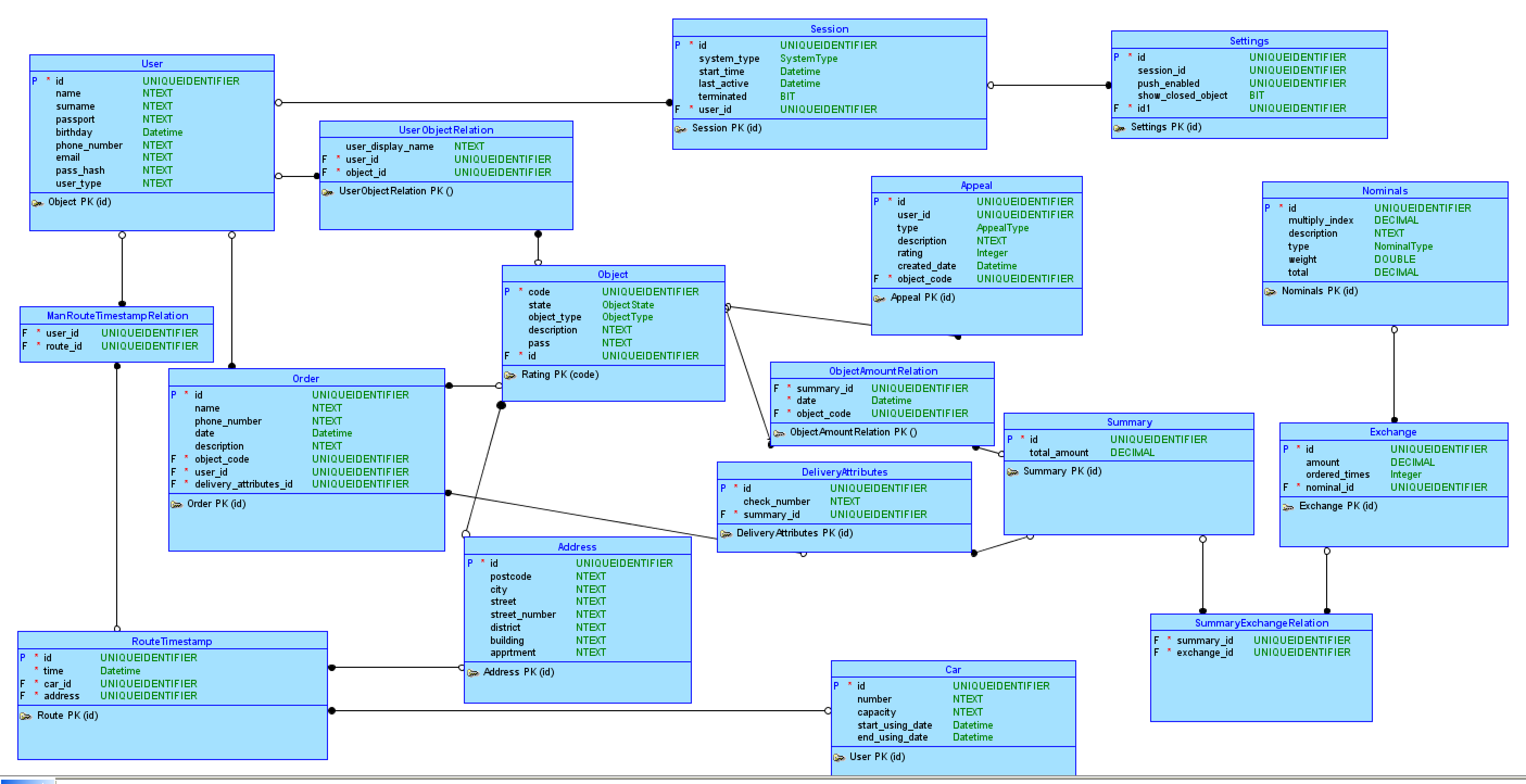


Рисунок 6 – Нормализованная логическая модель

# 15. Построение физической модели данных системы. Определение типов данных для атрибутов. Создание реляционной модели.

Для выполнения поставленной цели необходимо определить требуемые и минимально необходимые типы данных для различных атрибутов в схеме базы данных «Служба инкассации». Пример финальной схемы приведен на рисунке 7.

Для установления типов данных были введены собственные типы данных, пример которого – AppealType. Тип данных, который определят тип обращений. Представляет из себя перечисление на основе целочисленного типа данных, имеет следующие состояния – отзыв (review) и жалобу (claim). Также были выставлены все типы данных на соответствующих моделях. Рассмотрим в качестве примера модель пользователя.

Таблица 8 – Схема «User»

|  |  |
| --- | --- |
| Атрибут | Тип данных |
| id | uuid |
| name | varchar |
| surname | varchar |
| passport | varchar |
| birthday | datetime |
| phone\_number | varchar |
| email | varchar |
| pass\_hash | varchar |
| user\_type | UserType |

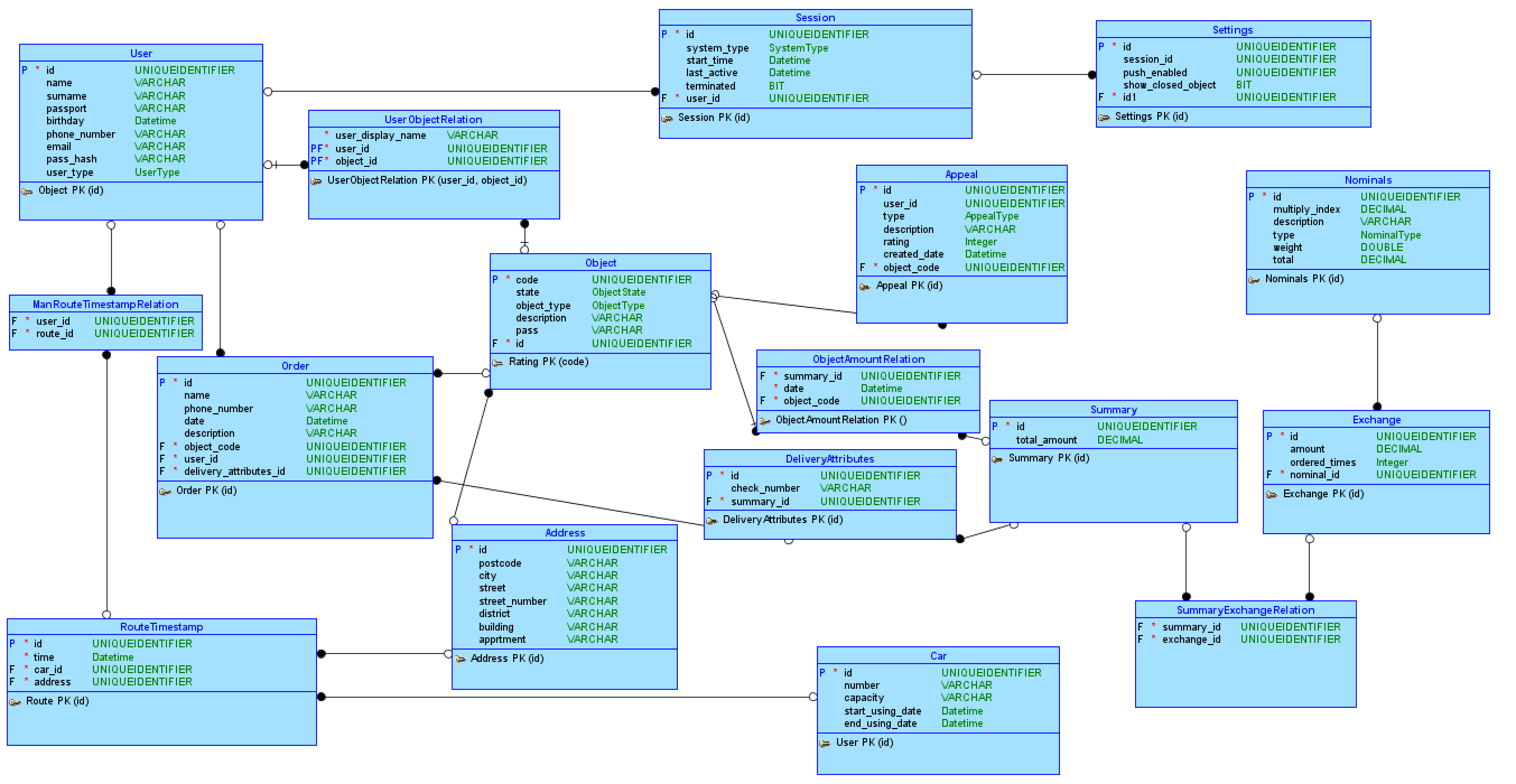


Рисунок7 – Финальная схема БД

# 16. Составление основных пунктов технического задания на проект

## 1.Бизнес требования

**1.1 Исходные данные**

Требуется разработать приложение, которое будет осуществлять учет и обслуживание клиентов службы инкассации.

**1.2 Возможности бизнеса**

С помощью этого приложения появиться возможность ускорения процесса регистрации клиентов в системе, упрощение процесса создания заявок. Отслеживание, учет и планирование маршрутов путем введения электронной системы упростятся, что позволит снизить нагрузку на персонал и уменьшить возможность ошибок.

**1.3 Бизнес цели**

**BO-1** Уменьшить время обработки заявок клиентов на 60%

**BO-2** Увеличить скорость построения маршрутов передвижения на 40%

**BO-3** Снижениевероятности ошибок при работе с данными для сотрудников

**BO-4** Внедрение системы оценок и отзывов для всех участников процесса

**BO-5** Добавление функциональности просмотра наиболее популярных форматов и количества размена

**BO-6** Увеличение производительности и эффективности кассово-инкассационных центров

**1.4 Критерии успеха**

**SM-1** Увеличение скорости заявки и снижение времени ожидания клиентов путем ввода электронной системы

**SM-2** Снижение нагрузки на персонал и улучшение его качества работы путем снижения количества ошибок

**SM-3** Скорость формирования отчетов и поиска информации увеличится на 75%

**1.5 Видение решения**

Информационная система «Служба инкассации» – это компьютерное приложение, которое используют клиенты службы инкассации для создания заявок на инкассацию, заявок на размен валюты с указанием времени и объекта инкассации. Сотрудники службы проводят подтверждение заявок, учет и регистрацию объектов инкассации. Также у сотрудников есть возможность просматривать отчеты по наиболее популярным разменам, по отзывам клиентов.

**1.6 Бизнес-риски**

**RI-1** Отсутствие опыта работы с подобными системами у сотрудников службы инкассации может первое время еще больше замедлить процесс работы

**RI-2** Имеющегося технического оборудования может не хватить для внедрения системы

## 2.Рамки и ограничения проекта

**2.1 Основные функции**

**FE-1** Создание заявки на инкассацию со стороны клиента

**FE-2** Отслеживание заявки

**FE-3** Регистрация клиентов в системе

**FE-4** Добавление объектов со стороны клиентов

**FE-5** Регистрация объектов со стороны сотрудников

**FE-6** Учет необходимого количества размена в машину инкассации

**FE-7** Учет максимального допустимого количества по весу в машине инкассации

**FE-8** Создание отзывов любым участником системы

**FE-9** Просмотр сгенерированных отчетов по отзывам

**FE-10** Генерация отчетов по наиболее популярным единицам размена

**FE-11** Прогноз необходимого денежного количества в объектах

**FE-12** Доступ к маршрутным картам для инкассаторов

**2.2 Состав первого и последующих выпусков**

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Выпуск 1 | Выпуск 2 | Выпуск 3 |
| FE-1 | Реализована полностью |  |  |
| FE-2 | Реализована полностью |  |  |
| FE-3 | Реализована полностью |  |  |
| FE-4 | Реализована полностью |  |  |
| FE-5 | Частично реализовано | Реализована полностью |  |
| FE-6 | Не реализована | Частично реализовано | Реализована полностью |
| FE-7 | Не реализована | Частично реализовано | Реализована полностью |
| FE-8 | Частично реализовано | Реализована полностью |  |
| FE-9 | Частично реализовано | Реализована полностью |  |
| FE-10 | Не реализовано | Частично реализовано | Реализована полностью |
| FE-11 | Не реализовано | Частично реализовано | Реализована полностью |
| FE-12 | Частично реализовано | Реализована полностью |  |

# Заключение

В процессе выполнения курсовой работы по определению и проектированию архитектуры приложения, определению распределенной системы и выбора правильной технологии было получено видение приложения, которое решает поставленную перед ним задачу, предоставляет достаточный набор функциональных и нефункциональных требований, грамотную реализацию модели базы данных через первоначальное проектирование приложения на уровне логического моделирования. Такой вид проектирования позволяет избегать отношения многие-ко-многим, грамотно избегать дублирования сущностей и данных.

У разработанного приложения хорошая перспектива внедрения в промышленные масштабы для существующих служб инкассации. Ключевым достоинством будет наличие открытости решения, которое позволит составить достаточную конкуренцию закрытым решениям, находящимся внутри компаний.