МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

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

**МЕХАНИКИ И ОПТИКИ»**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**«Разработка распределенной системы»**

по дисциплине: Разработка и внедрение распределенных систем.

Студент Кислюк Игорь Витальевич

Факультет Инфокоммуникационные технологии

Кафедра Программных систем

Группа К4120

Направление (специальность) 11.04.02 Программное обеспечение в инфокоммуникациях

Проверил: Иванов Сергей Евгеньевич, доцент, к.ф.-м.н.

Санкт-Петербург

2018 г.

# 1. 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 1, указаны основные действующие лица системы, порождающие и принимающие потоки данных.

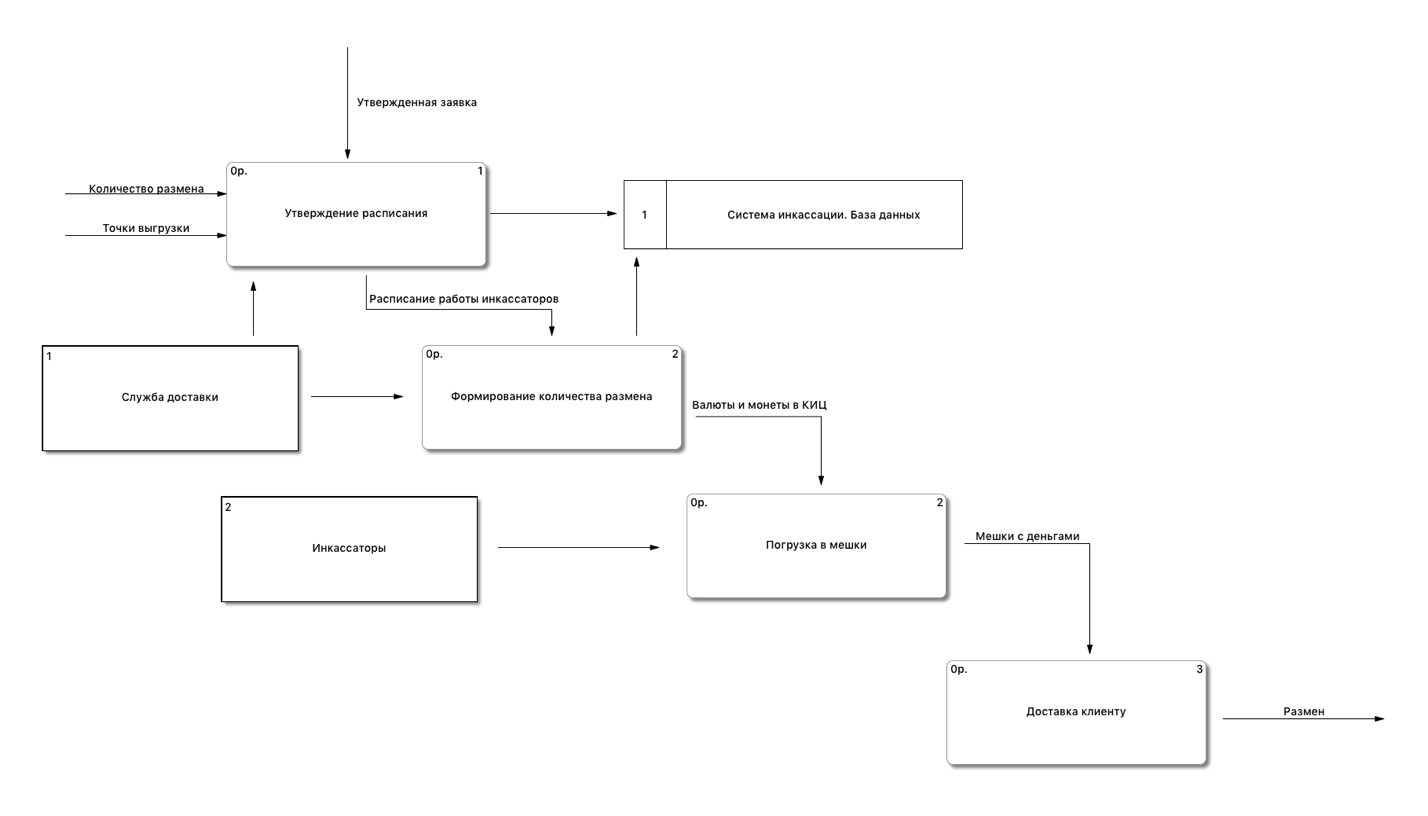
****

Рисунок 1 – Диаграмма потоков данных

# 2. Определение структуры информационных потоков

Структура информационных потоков в информационной системе службы инкассации состоит из следующих элементов, представленных в таблице 1.

Таблица 1 – 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 |

# 3. Разработка функциональной архитектуры системы

Разработка функциональной архитектуры данной системы может быть представлена графически. Для этого определения выступает инфологическая модель данных, представленная на рисунке 2.

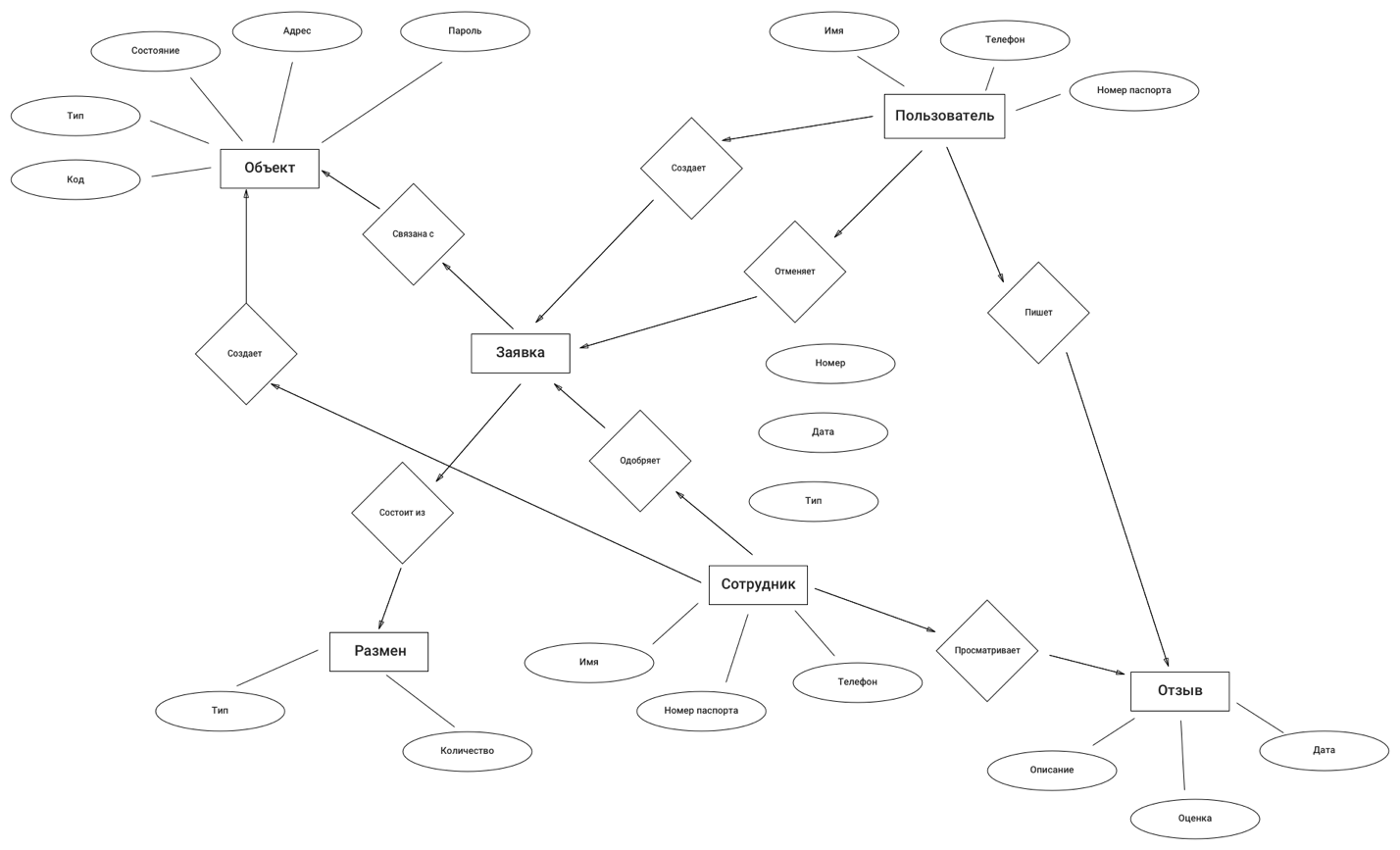


Рисунок 2 – Инфологическая модель

# 4. 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.

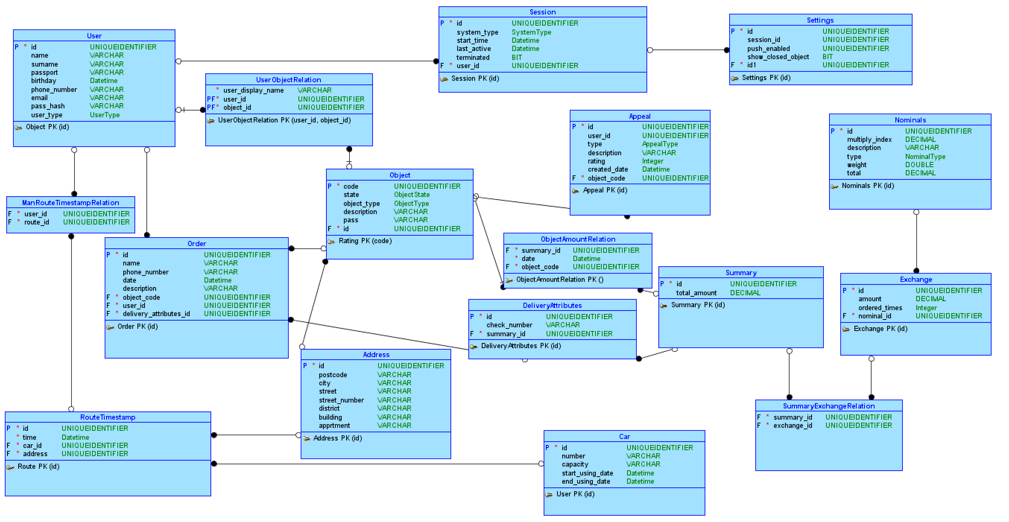
*Недостатки:*

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

Определив технологии разработки необходимо приступать к разработке логической модели данных.

# 5. Разработка логической модели данных информационной системы.

Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения. На рисунке 3 показана возможная логическая модель данных информационной системы службы инкассации.

Рисунок 3 – Логическая модель данных

# 6. 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 4.

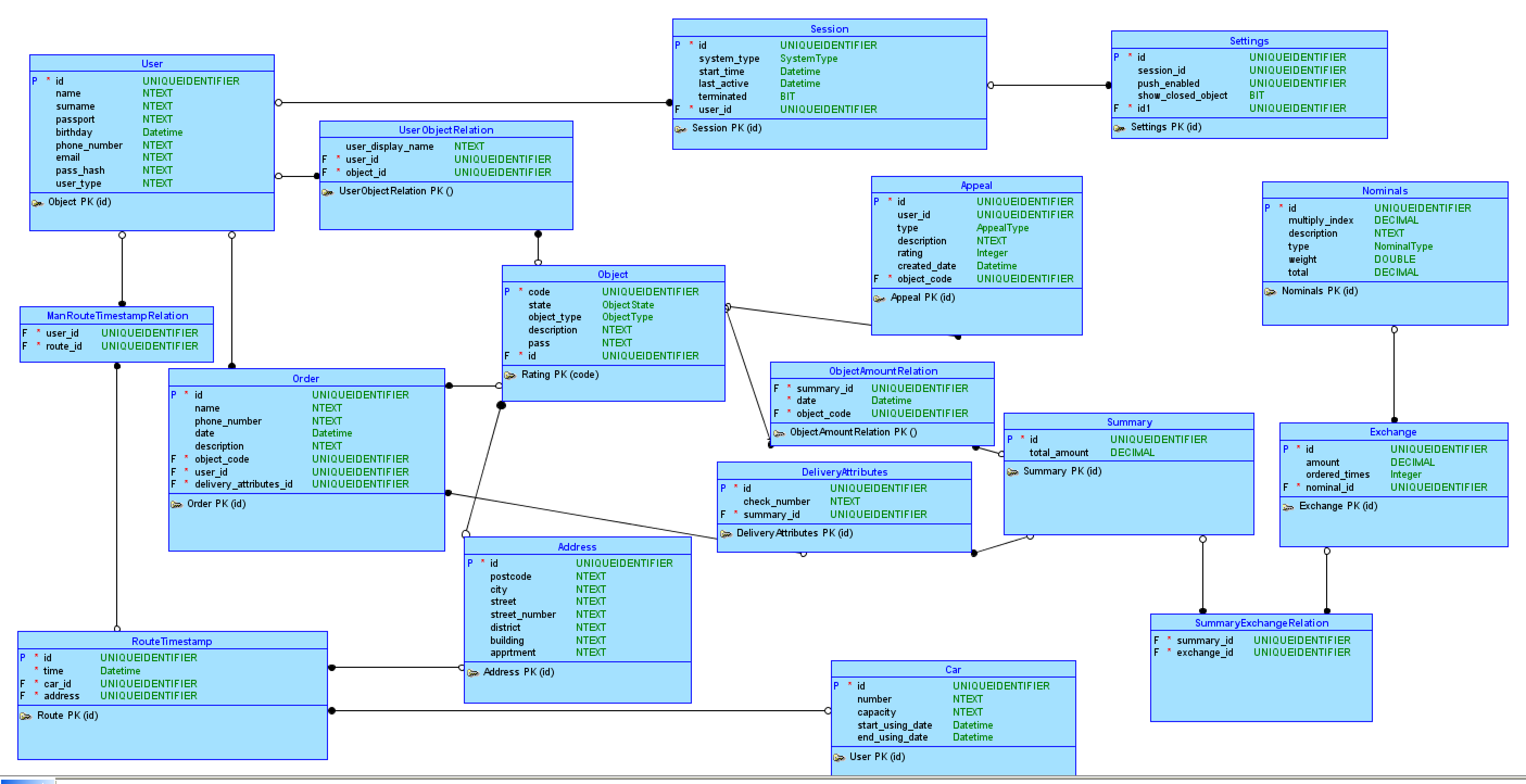


Рисунок 4 – Нормализованная логическая модель

# 7. 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 5.

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 2 – 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 |

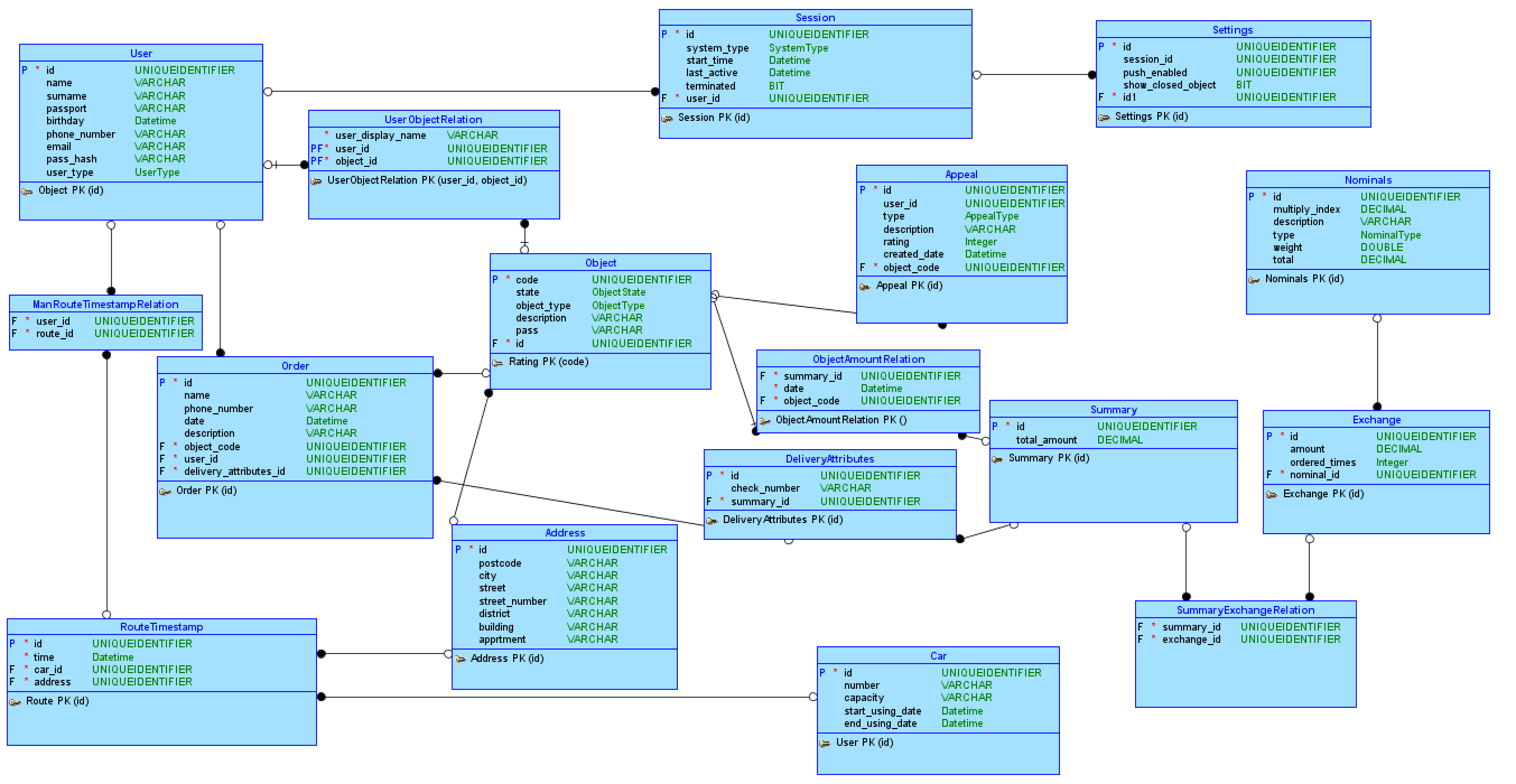


Рисунок *5* – Финальная схема БД

# 8. 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 лабораторной работы по разработке распределенной системы и выбору правильной технологии было получено приложения, которое решает поставленную перед ним задачу, предоставляет грамотную реализацию модели базы данных на основе правильного проектирования.

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.