

РЕФЕРАТ

Пояснительная записка 84 с, 38 рисунков, 3 таблицы, 20 источников, 2 приложения.

Графическая часть: 19 слайдов презентации PowerPoint.

ДОРОЖНЫЕ ЗНАКИ, КОРИЧНЕВЫЕ ЗНАКИ, ТУРИСТИЧЕСКИЕ ЗНАКИ, ПРОЕКТИРОВАНИЕ ЗНАКОВ, УКАЗАТЕЛЬ НАПРАВЛЕНИЙ, РЕДАКТОР ЗНАКОВ, ЗНАКИ ИНДИВИДУАЛЬНОГО ПРОЕКТИРОВАНИЯ, ДОРОЖНОЕ ДВИЖЕНИЕ.

Во время курсового проектирования разработан плагин, позволяющий осуществлять дислокацию геообъектов «Коричневые знаки» для информационной системы ITSGIS. Программа позволяет создать тематический слой геообъектов «Коричневые знаки» на электронной карте города, а также предоставляет возможность добавления, выделения, редактирования и удаления геообъектов, группировку геообъектов по тематическим слоям. В системе имеется возможность сохранения данных о коричневых знаках и их характеристиках в базу данных. Данные о лестницах (тип, размер, место расположения и т.д.) хранятся в файлах типа *.sql.

Программа написана на языке C# в среде Visual Studio 2019 и функционирует под управлением операционной системы Windows 10 и выше. Для работы с данными используется СУБД PostgreSQL 9.1 с расширением PostGIS 1.5.

СОДЕРЖАНИЕ

Введение.....	18
1.1.1 Классификация дорожных знаков	20
1.1.2 Знаки индивидуального проектирования	23
1.1.3 Коричневые знаки	24
1.2 Описание систем-аналогов.....	25
1.2.1 Яндекс.Карты.....	26
1.2.2 2ГИС	27
1.2.3 Google Maps	29
1.2.4 Сравнение систем-аналогов по различным критериям.....	30
1.3 Постановка задачи	32
2 Проектирование системы	35
2.1 Структурная схема системы	35
2.2 Общие сведения об ITSGIS	35
2.3 Разработка прототипа интерфейса пользователя системы.....	40
2.4 Разработка информационно-логического проекта системы.....	48
2.4.1 Язык UML	48
2.4.2 Диаграмма вариантов использования	48
2.4.3 Сценарии	50
2.4.4 Диаграмма классов.....	54
2.4.5 Диаграмма состояний	57
2.4.6 Диаграмма деятельности	57
2.4.7 Диаграмма последовательности	61
2.5 Логическая модель данных	61
2.6 Описание комплекса программных средств	65
2.6.1 Описания языка программирования и среды разработки.....	66
2.6.2 Описание операционной системы	66
2.6.3 Описание среды разработки.....	67
2.6.4 Описание системы управления базами данных	67

3 Реализация системы	69
3.1 Разработка и описание интерфейса пользователя	69
3.1.1 Описание тестового примера.....	72
3.2 Диаграммы реализации	73
3.2.1 Диаграмма компонентов	73
3.2.2 Диаграмма развертывания.....	74
3.2.3 Диаграмма классов.....	75
3.3 Выбор и обоснование комплекса технических средств.....	76
3.3.1 Расчёт объёма занимаемой памяти клиентской части системы.	76
3.3.2 Минимальные требования, предъявляемые к системе изображения расстояния между компонентами знака	77
Заключение	78
Список использованных источников	79
Приложение А Руководство пользователя	81
Приложение Б Листинг модулей программы.....	86

Введение

Правила дорожного движения (ПДД) – свод правил, которые регулируют права и обязанности водителей и пешеходов. В рамках ПДД существуют дорожные знаки, регулирующие движение. Знаки бывают запрещающие, предупреждающие, предписывающие и другие. Немаловажными являются информационные знаки, в частности коричневые (туристические). Коричневые дорожные знаки предназначены для того, чтобы помочь туристам ориентироваться в незнакомой местности.

Органы местного самоуправления (муниципалитеты) ответственны за установку и обслуживание дорожных знаков, в том числе туристических. У муниципалитетов существует потребность в организации информации обо всех дорожных знаках своей местности – такую потребность удовлетворяют геоинформационные системы.

Во время курсового проектирования необходимо разработать плагин дислокации геообъектов коричневых (туристических) дорожных знаков на электронной карте в среде интеллектуальной транспортной геоинформационной системы ITSGIS, с помощью которой можно выделять, добавлять, наполнять редактировать, копировать, удалять этот геообъект на электронной карте и в базе данных.

При проектировании системы будут использоваться методологии структурного проектирования систем и ООАП (Object-Oriented Analysis/Design), в основу которой положена объектно-ориентированная методология представления предметной области в виде объектов, а также язык моделирования UML (Unified Modeling Language), который является стандартным инструментом для разработки «чертежей» программного обеспечения.

1 Описание и анализ предметной области

Дорога, по которой едет автомобиль в большом городе – это, как правило, многополосный путь с большим количеством участников дорожного движения. Участники дорожного движения – это водители транспортных средств, пешеходы, пассажиры транспортных средств. Дорожное движение регулируется сводом правил – Правила дорожного движения (ПДД).

ПДД – это свод правил, регулирующих обязанности водителей транспортных средств и пешеходов, а также технические требования, предъявляемые к транспортным средствам для обеспечения безопасности.

В частности, для регулирования дорожного движения применяются знаки. Некоторые дорожные знаки используются для навигации, например, туристические (коричневые) [1].

1.1 Описание предметной области

Дорожный знак – техническое средство безопасности дорожного движения, стандартизованный графический рисунок, устанавливаемый у дороги для сообщения определённой информации участникам дорожного движения [1].

Первые в мире дорожные знаки были утверждены на международной конференции автомобилистов в 1909 году. В числе участников были и представители Российской Империи. Дорожных знаков того времени было четыре и все круглой формы: «перекресток», «шлагбаум», «двойной поворот», «препятствие в виде насыпи и канавы».

Новые дорожные знаки и сигналы были официально приняты уже в СССР 1 декабря 1927 года. В 1933 году их количество увеличилось до двадцати трёх, и они получили привычные сегодня формы и цвет, а также впервые были разделены на три категории: «предупреждающие», «воспрещающие» и «указательные». Следующие изменения и дополнения, касающиеся дорожных знаков, были приняты 1 января 1961 года, после

присоединения СССР в 1959 году к Международной конвенции по дорогам и моторному транспорту 1949 года.

В 1968 году в Вене была принята Конвенция о дорожном движении и дорожных знаках и сигналах. Среди стран, принявших Венскую конвенцию, был и СССР. Новые правила дорожного движения, а также дорожные знаки, принятые этой конвенцией, вступили в силу в СССР в 1973 году. В дальнейшем изменения и дополнения в правила дорожного движения, дорожные знаки и сигналы вносились в последующие годы, вплоть до настоящего времени [2].

Общие принципы дорожных знаков и обозначений совпадают во многих странах, но сами знаки могут достаточно сильно различаться от страны к стране. Большинство систем дорожных знаков мира можно свести к двум системам, европейской и англосаксонской, с некоторыми местными дополнениями и различиями. В Европе, России и многих других странах мира действует Венская конвенция о дорожных знаках и сигналах [1].

Далее, рассмотрим классификацию дорожных знаков.

1.1.1 Классификация дорожных знаков

Согласно Постановлению Правительства РФ «О Правилах дорожного движения» (и Венской конвенции дорожные) знаки делятся на восемь категорий [1].

«Предупреждающие знаки» представлены на рисунке 1. Имеют треугольную форму, белый фон с красной окантовкой и чёрные рисунки. Предупреждающие знаки информируют водителей о приближении к опасному участку дороги, движение по которому требует принятия мер, соответствующих обстановке.

«Знаки приоритета» представлены на рисунке 2. Форма намеренно разная, чтобы исключить ошибочные предположения о значении знака даже при взгляде с тыльной стороны. Знаки приоритета устанавливают

очередность проезда перекрестков, пересечений проезжих частей или узких участков дороги.



Рисунок 1 – Примеры предупреждающих знаков

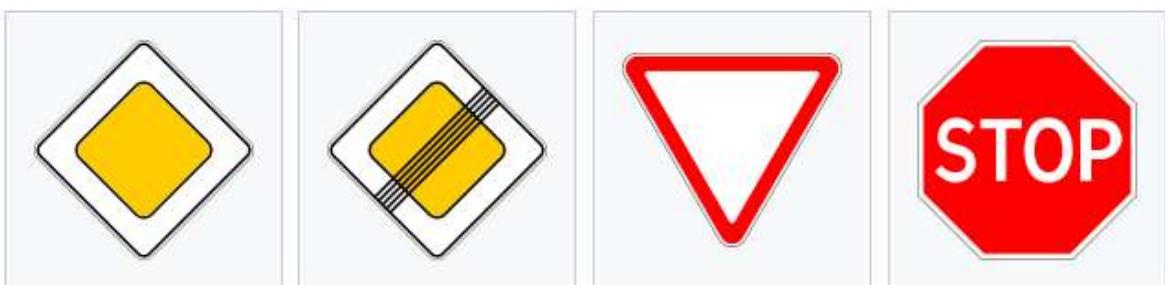


Рисунок 2 – Примеры знаков приоритета

«Запрещающие знаки» представлены на рисунке 3. Такие знаки имеют круглую форму, белый фон и чёрные рисунки. Вводят или отменяют определённые ограничения в движении. Действие знаков начинается непосредственно с того места, где они установлены, на ту сторону проезжей части, на которой они установлены и распространяют своё действие до первого перекрёстка или до конца населённого пункта.

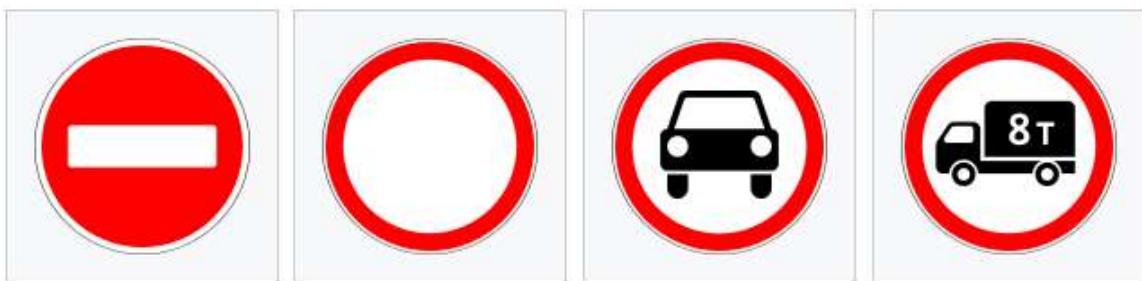


Рисунок 3 – Примеры запрещающие знаки

«Предписывающие знаки» представлены на рисунке 4. Такие знаки имеют круглую форму, синий фон и белые рисунки. Предписывают участникам дорожного движения определённые действия, например, направление поворотов.

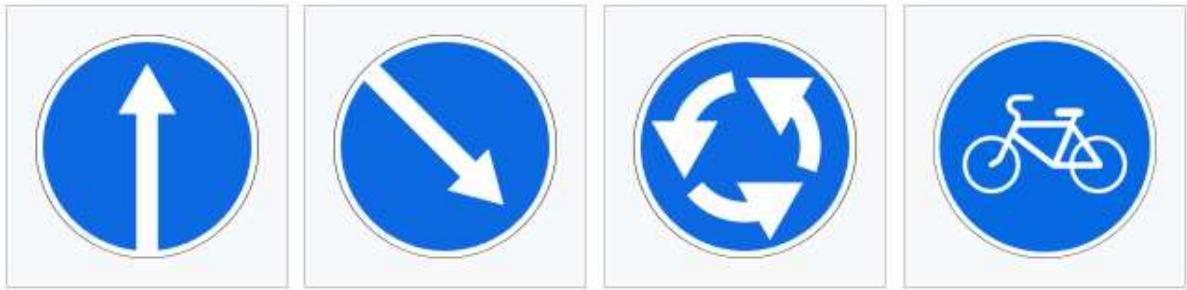


Рисунок 4 – Примеры предписывающих знаков

«Знаки особых предписаний» представлены на рисунке 5. Такие знаки имеют прямоугольную форму и, как правило, синий фон со светлыми символами или надписями. Знаки особых предписаний вводят или отменяют определенные режимы движения.



Рисунок 5 – Примеры знаков особых предписаний

«Информационные знаки» представлены на рисунке 6. Такие знаки имеют квадратную или прямоугольную форму, цвет фона обычно синий (реже – зелёный), цвет рисунков обычно белый. Информационные знаки информируют о расположении населенных пунктов и других объектов, а также об установленных или о рекомендуемых режимах движения.

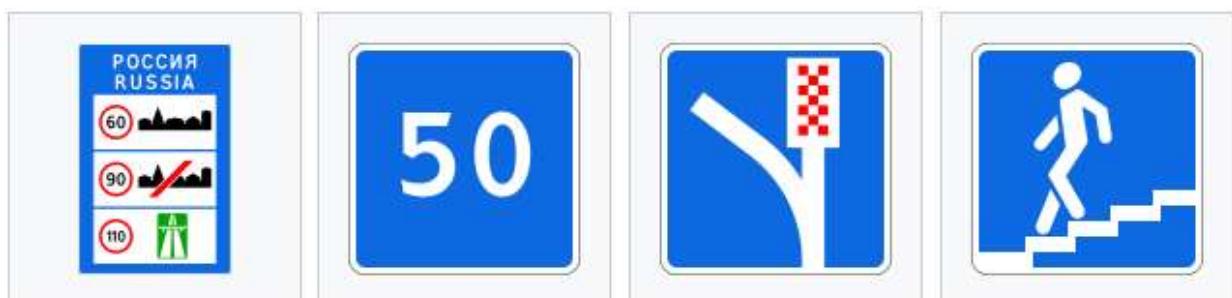


Рисунок 6 – Примеры информационных знаков

«Знаки сервиса» представлены на рисунке 7. Такие знаки имеют прямоугольную форму, белый фон с синей окантовкой и чёрные рисунки.

Информируют участников дорожного движения о разных услугах: автозаправочных станциях, гостиницах, кемпингах и др.

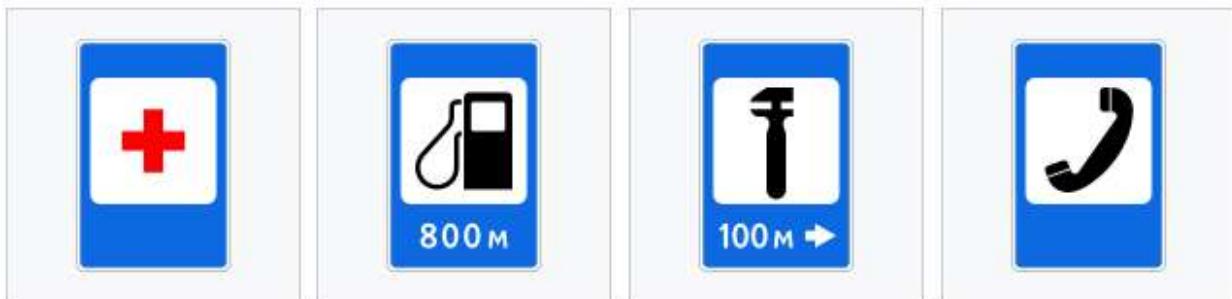


Рисунок 7 – Примеры знаков сервиса

«Знаки дополнительной информации (таблички)» представлены на рисунке 8. Такие знаки имеют прямоугольную форму, белый фон с чёрной окантовкой и чёрными рисунками. Являются дополнительными к знакам вышеперечисленных категорий, отдельно не используются. Уточняют действия основных знаков по времени (например, только по будним дням) или распространяя их только на определённые категории транспортных средств (например, только для грузовиков), или предоставляют другую дополнительную информацию.

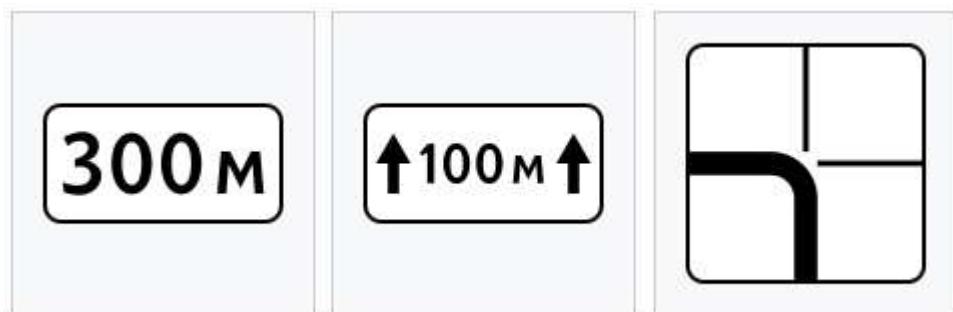


Рисунок 8 – Примеры знаков дополнительной информации

1.1.2 Знаки индивидуального проектирования

Знаки индивидуального проектирования (ЗИП) относятся категории информационных знаков. ЗИП информируют о расположении населенных пунктов, объектов туризма, схем, различных придорожных объектов или элементов дорожной сети, к примеру, улиц, площадей, мостов и т.д. Знаки

индивидуального проектирования позволяют лучше ориентироваться в маршрутах. Пример ЗИП представлен на рисунке 9.

Знаки индивидуального проектирования называются так, потому что для каждого знака проект просчитывается отдельно, в зависимости от размеров и содержания. Существует государственный стандарты на размер букв и расстояние между ними, на цвет фона и другие особенности таких знаков [3].

Исходя из вышесказанного, знаки индивидуального проектирования могут иметь различные размеры. Места размещения для таких знаков тоже весьма варьируются. Они могут находиться на дорогах, автомагистралях, в населенном пункте и за его чертой.

Характеристики ЗИП представлены в приложении к исходным данным к геообъекту «знак индивидуального проектирования».



Рисунок 9 – Пример знака индивидуального проектирования

1.1.3 Коричневые знаки

Коричневые знаки – знаки индивидуального проектирования под номером ПДД 6.11 («Наименование объекта»), коричневого цвета, обозначают наименование туристического объекта, иного чем населенный

пункт (река, озеро, перевал, достопримечательность и т.п.). Пример коричневого знака представлен на рисунке 10.

Знаки имеют соответствующего цвета фон, надписи выполняются белым цветом, возможно изображение направления (стрелок) и пиктограмма обозначающая тип туристического объекта.



Рисунок 10 – Пример коричневого знака

ГОСТ Р 52290-2004 различает пять типов коричневых (туристических) знаков:

- «музей, усадьба, художественная галерея»;
- «природный заповедник, национальный парк, ботанический сад»;
- «исторический памятник, культовое сооружение – памятник архитектуры»;
- «памятник архитектуры»;
- «спортивный объект»;

Соответствующие этим типам пиктограммы представлены в таком же порядке на рисунке 11.

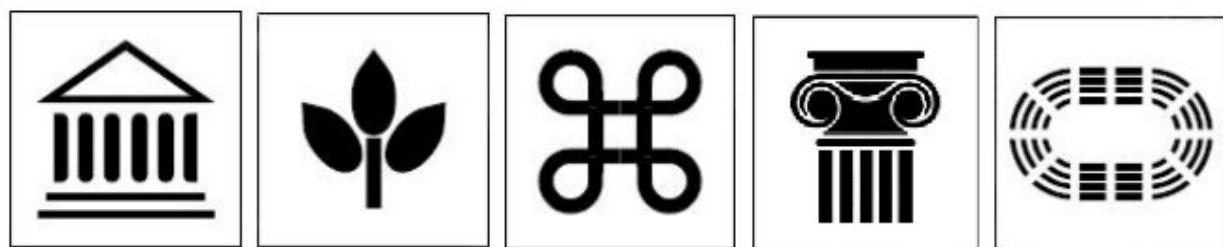


Рисунок 11 – Пиктограммы типов коричневых знаков

1.2 Описание систем-аналогов

Существуют различные геоинформационные системы, которые позволяют выделять и добавлять геообъекты, наполнять и редактировать картографическую информацию.

В качестве примеров, рассмотрим такие геоинформационные системы как «2ГИС», «Яндекс.Карты» и Google Maps.

1.2.1 Яндекс.Карты

«Яндекс.Карты» – геоинформационная система компании «Яндекс». По состоянию на октябрь 2020 года российская ежемесячная аудитория системы составляла 38,4 млн. пользователей [4].

Карты доступны в четырёх вариантах: схемы, спутниковые снимки, гибрид (спутниковые снимки с надписями и условными обозначениями) и «Народная карта» (карта, на которой пользователи сами рисуют и уточняют карту). Набор возможностей по работе с картами достаточно обширен, но в наиболее полном объёме доступен лишь для ограниченного числа городов.

На картах и слоях карт отсутствуют геометки, обозначающие дорожные знаки, в том числе коричневые знаки. В «Народной карте» имеются геометки, обозначающие условия движения дорог, на рисунки 12 представлен пример такой геометки (обозначается белым кругом в круглой жёлтой рамке).

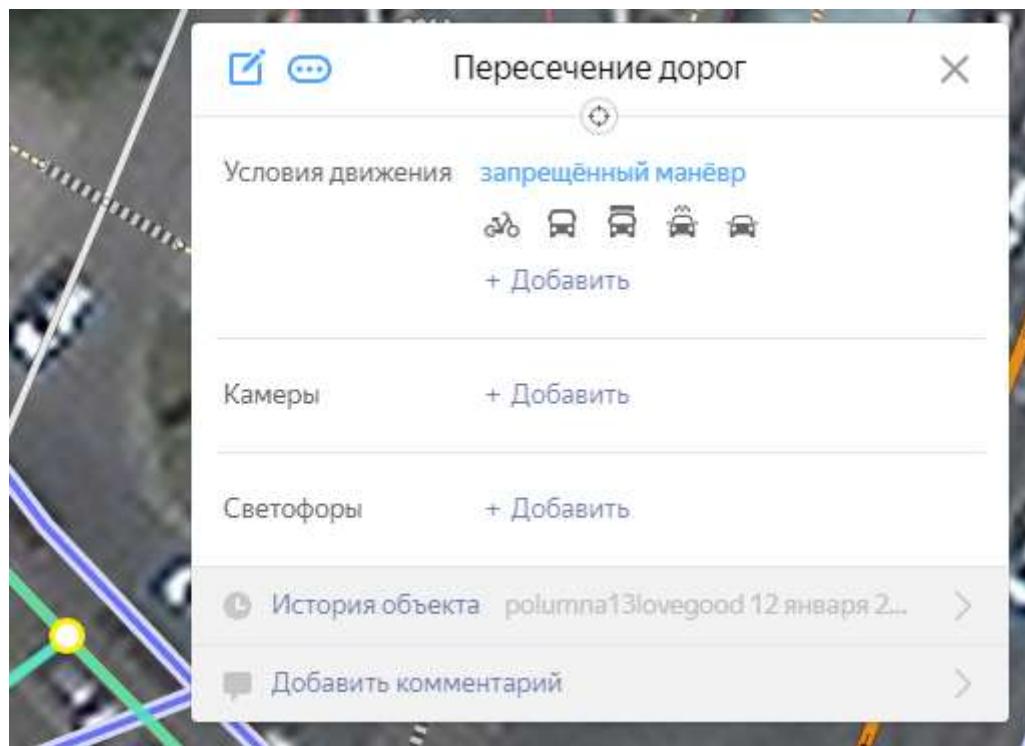


Рисунок 12 – Пример геометки, обозначающей условия движения дорог

При нажатии на геометку, в правом верхнем углу карты появляется окно с информацией о геометке. Окно содержит название, условие движения и пиктограммы транспортных средств, на которые действует условие движения. В этом же окне можно тут же отредактировать информацию об этой геометке: добавить и удалить пиктограммы транспортных средств, камер и светофоров, редактировать местоположение геометки. Из этого окна можно перейти в окно истории правок информации об этой геометки «История объекта» и в окно обсуждения геометки «Добавить комментарий».

1.2.2 2ГИС

«2ГИС» – геоинформационная система одноимённой компании. По данным компании на 11 июня 2020 года, ежемесячная аудитория сервиса составляет около 50 миллионов пользователей [5].

«2ГИС» включает в себя карту, справочник организаций, поиск проезда на общественном и личном транспорте, линейку для измерения расстояний, отображение пробок в некоторых городах. Работает с API собственной разработки.

На картах и слоях карт отсутствуют геометки, обозначающие дорожные знаки. Имеются геометки, обозначающие туристические объекты, например, памятники. Пример такой геометки представлен на рисунке 13.

При нажатии на геометку, слева на карте выдвигается панель поиска, на которой отображена информация о геометке и доступны две вкладки «Инфо» и «Отзывы».

На вкладке «Инфо» содержится краткое описание и фотографии геообъекта. На вкладке «Отзывы» находятся отзывы о геообъекте,ставленные пользователями «2ГИС», также возможно написать собственный отзыв. Пример вкладки «Отзывы» представлен на рисунке 14.

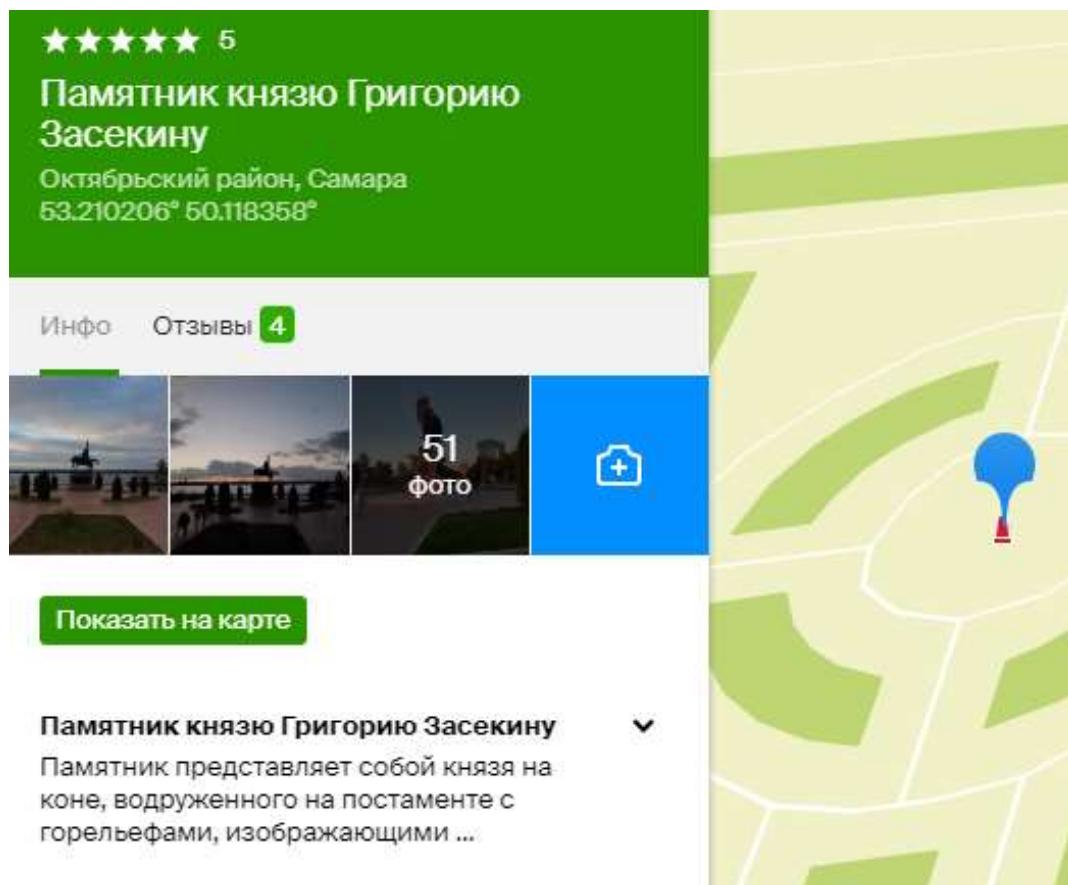


Рисунок 13 – Пример геометки, обозначающей туристический объект

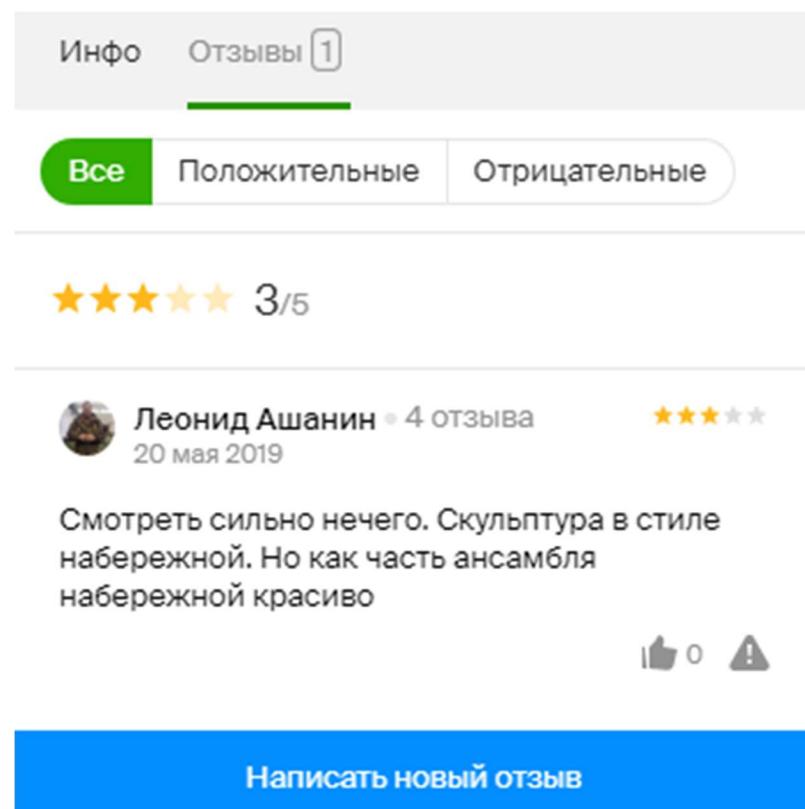


Рисунок 14 – Пример вкладки «Отзывы»

1.2.3 Google Maps

Google Maps – геоинформационная система компании Google. Компания не раскрывает информацию ежемесячной аудитории сервиса Google Maps [6].

Карты доступны в трёх режимах: карта (схема), спутник и рельеф. Набор возможностей по работе с картами ограничен для российских пользователей, включая пользователей крупных городов: отсутствует подробная информация об организациях и услугах городов.

На картах и слоях карт отсутствуют геометки, обозначающие дорожные знаки. Имеются геометки обозначающие туристические объекты, например, памятники, обозначаются бирюзовым маркером с изображением фотоаппарата. Пример такой геометки представлен на рисунке 15.

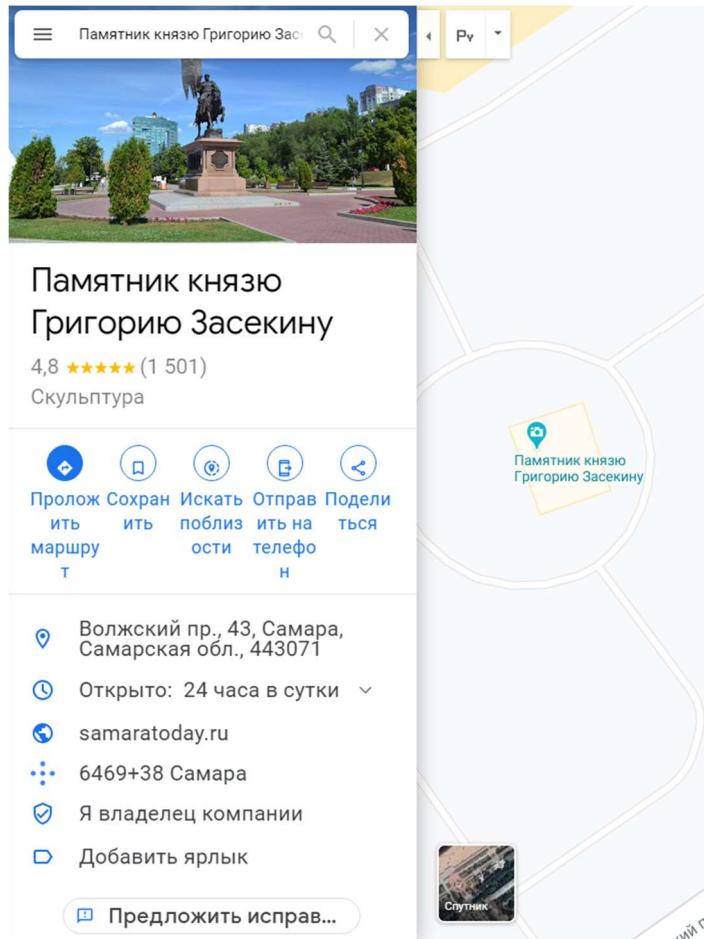


Рисунок 15 – Пример геометки, обозначающей туристический объект

При нажатии на геометку, слева на карте выдвигается панель поиска, на которой отображена информация о геообъекте: название, адрес, время работы, плюс-код (код, основанный на географических координатах объекта, обозначает местоположение), график посещаемости, фотографии, отзывы.

1.2.4 Сравнение систем-аналогов по различным критериям

В таблице 1 приведено сравнение сервисов по различным критериям [7-9].

Таблица 1 – Сравнение геоинформационных систем по различным критериям

Критерий	Яндекс.Карты	2ГИС	Google Maps
1	2	3	4
Покрытие	Россия, мир	Россия, мир	Россия, мир
Детализация на уровне здания	Имеется	Имеется	Имеется
Возможность использования онлайн	Имеется	Имеется	Имеется
Редактирование карт	Сервис «Народная карта», сообщение об ошибках	Сервис 2GIS Business Account (редактирование информации о бизнесе пользователя сервиса), сообщение об ошибках	Сервис «Google Мой бизнес» (редактирование информации о бизнесе пользователя сервиса), сообщение об ошибках

Продолжение таблицы 1

1	2	3	4
Типы карт	Схема, спутник, гибрид	Карта	Карта (схема), спутник, рельеф
Отображение пробок в крупных городах	Имеется	Имеется	Имеется
Панорамный просмотр улиц	Имеется	Отсутствует	Имеется (Google Street View)
Голосовой ввод на русском языке	Имеется	Имеется	Имеется
Построение маршрута	Автомобиль, общественный транспорт, пеший, велосипед, такси	Автомобиль, общественный транспорт, пеший, такси	Автомобиль, общественный транспорт, пеший, велосипед, самолёт
Справочная информация	Подробная информация об организациях	Подробная информация об организациях	Подробная информация об организациях
Возможность оставить отзыв и оценить организацию	Имеется	Имеется	Имеется
Интерфейс и использование	Веб, Android, iOS	Веб, Android, iOS	Веб, Android, iOS

1.3 Постановка задачи

Во время курсового проектирования необходимо разработать плагин «Коричневые знаки» для геоинформационной системы ITSGIS, позволяющий добавлять, редактировать и просматривать информацию о коричневых знаках и их положении на карте.

В системе должны быть предусмотрены две роли пользователя – администратор и обычный пользователь. После авторизации пользователя (ввода логина и пароля) система должна проверить учётную запись и настроить интерфейс на заданную роль. После успешной авторизации вне зависимости от выбранной роли пользователь сможет просматривать электронную карту и информацию по выбранному геообъекту «ЗИП коричневый знак».

В режиме администратора пользователь будет осуществлять взаимодействие с электронной картой: создавать и редактировать геообъекты «Коричневые знаки» на карте. В случае создания нового геообъекта администратору необходимо будет привязать геообъект к карте и заполнить необходимую информацию о нём.

При создании геообъекта «Коричневый знак» администратор должен будет определить тип знака («музей, усадьба, художественная галерея», «природный заповедник, национальный парк, ботанический сад», «исторический памятник, культовое сооружение – памятник архитектуры», «памятник архитектуры», «спортивный объект», нестандартный), задать надпись, направление (юг, север, запад, восток, юго-запад, юго-восток, северо-запад, северо-восток;), высоту букв (75, 100 или 150 мм; 150 или 200 мм; 200 или 300 мм; 300 или 400 мм; 400 или 500 мм) и расстояния между элементами знака, указать размеры знака (длина в мм, ширина в мм, вертикальные и горизонтальные размеры между элементами в мм), выбрать статус знака (установлен, требуется, демонтировать, ремонт, временный), задать собственника знака (название, почтовый адрес, контактный телефон,

веб-сайт), задать коллекцию фотографий знака, задать высоту, толщину и количество опор знака (одна, две или три).

После успешного создания геообъекта в базе данных добавятся все необходимые записи о коричневом знаке, а также система отрисует на карте изображение созданного геообъекта. Администратор должен иметь возможность отредактировать любые характеристики знака.

Помимо создания и редактирования геообъекта, администратор сможет удалять, перемещать, поворачивать и выбранный геообъект.

Вне зависимости от роли пользователь сможет получить и скачать сводную ведомость по всем коричневым знакам.

Информация о коричневых знаках будет храниться в базе данных, структура которой будет определена в процессе проектирования.

В режиме обычного пользователя будут система предоставит пользователю возможность просмотра информации по выбранному геообъекту коричневого знака.

Таким образом, система должна выполнять следующие функции:

1 Функции, реализуемые системой:

- выполнение разграничений прав доступа (администратор, пользователь);
- авторизация пользователя в системе (ввод логина/пароля);
- аутентификация пользователя в системе;
- настройка интерфейса пользователя на заданную роль;
- синхронизация базы данных и электронной карты: загрузка карты из базы данных, сохранение изменений в базу данных;
- просмотр электронной карты по тематическому слою коричневых знаков.

2 Функции работы с электронной картой:

- создание, редактирование и удаление тематического слоя геообъектов «ЗИП коричневый знак»;

- отображение условного обозначения геообъекта «ЗИП коричневый знак»;
- отображение информации о геообъекте «ЗИП коричневый знак».

3 Функции работы с базой данных:

- получение сводной ведомости и экспорт её во внешний файл.

4 Функции работы с геообъектом «ЗИП коричневый знак»:

- визуализация геообъекта на карте;
- выделение, добавление нового геообъекта, наполнение семантики геообъекта, редактирование, удаление геообъекта;
- перемещение, поворот геообъектов на карте;
- создание индивидуальных данных и сводных ведомостей геообъектов;
- отображение семантической информации и статуса геообъекта.

2 Проектирование системы

2.1 Структурная схема системы

При построении структурной схемы программной системы она по функциональному признаку разделяется на основные подсистемы, между ними указываются информационные связи и/или связи по управлению, описывается основное назначение подсистем.

Система должна представлять собой совокупность элементов (объектов, субъектов), находящихся между собой в определённой зависимости и составляющих некоторое единство (целостность), направленное на достижение определенной цели. Система может являться элементом другой системы более высокого порядка (надсистема) и включать в себя системы более низкого порядка (подсистемы). То есть систему можно рассматривать как набор подсистем, организованных для достижения определенной цели и описанных с помощью набора моделей (возможно, с различных точек зрения), а подсистему – как группу элементов, часть которых составляет спецификацию поведения, представленного другими ее составляющими [10].

2.2 Общие сведения об ITSGIS

ITSGIS – это геоинформационная система (ГИС) с многослойной электронной картой города, обеспечивающая работу с различными геообъектами городской инфраструктуры (дома, дороги, дорожные знаки, светофоры, световые опоры, закрепленные территории, остановки общественного транспорта, транспортные маршруты и др.), специализированными геообъектами (ДТП, места концентрации ДТП, места работ, ведущихся на улично-дорожной сети, и др.) [11].

ITSGIS позволяет:

- отображать карты распространённых форматов;
- редактировать карту с помощью базовых графических примитивов;

- гибко настраивать пользовательский интерфейс;
- разрабатывать разнообразные модули («плагины»), расширяющие систему.

В системе организовано разграничение прав доступа пользователей на основе георолей: одним пользователям разрешён только просмотр информации, тогда как другие обладают правом модификации данных, причем права определяются как с учётом слоя доступа, так и области на карте.

ITSGIS построена на трёхзвенной клиент-серверной архитектуре (структурная схема представлена на рисунке 16):

- сервер баз данных – обеспечивает хранение данных, выполнение геопространственных операций, реализуется с использованием СУБД PostgreSQL, MS SQL Server, Oracle, MySQL;
- сервер приложений – предназначен для обеспечения многопользовательской работы, кеширования, защиты, шифрования, сжатия геоданных, выполнения ресурсоемких операций;
- клиенты – представляют пользователю доступ к функциям системы, реализованы как настольное приложение, веб-сайт.

Дополнительные программные модули (плагины) расширяют функциональность системы и позволяют работать с различными геообъектами – точечными, линейными и полигональными геометриями на электронной карте с прикрепленной к ним семантической информацией.

В состав любого плагина должны входить следующие подсистемы:

- подсистема проверки сервера, обеспечивающей синхронизацию приложения с сервером;
- подсистема авторизации, которая определяет роли пользователей;
- подсистема доступа к серверу;
- файловая подсистема;
- подсистема кэширования данных, делающая локальный кэш;
- подсистема настроек;

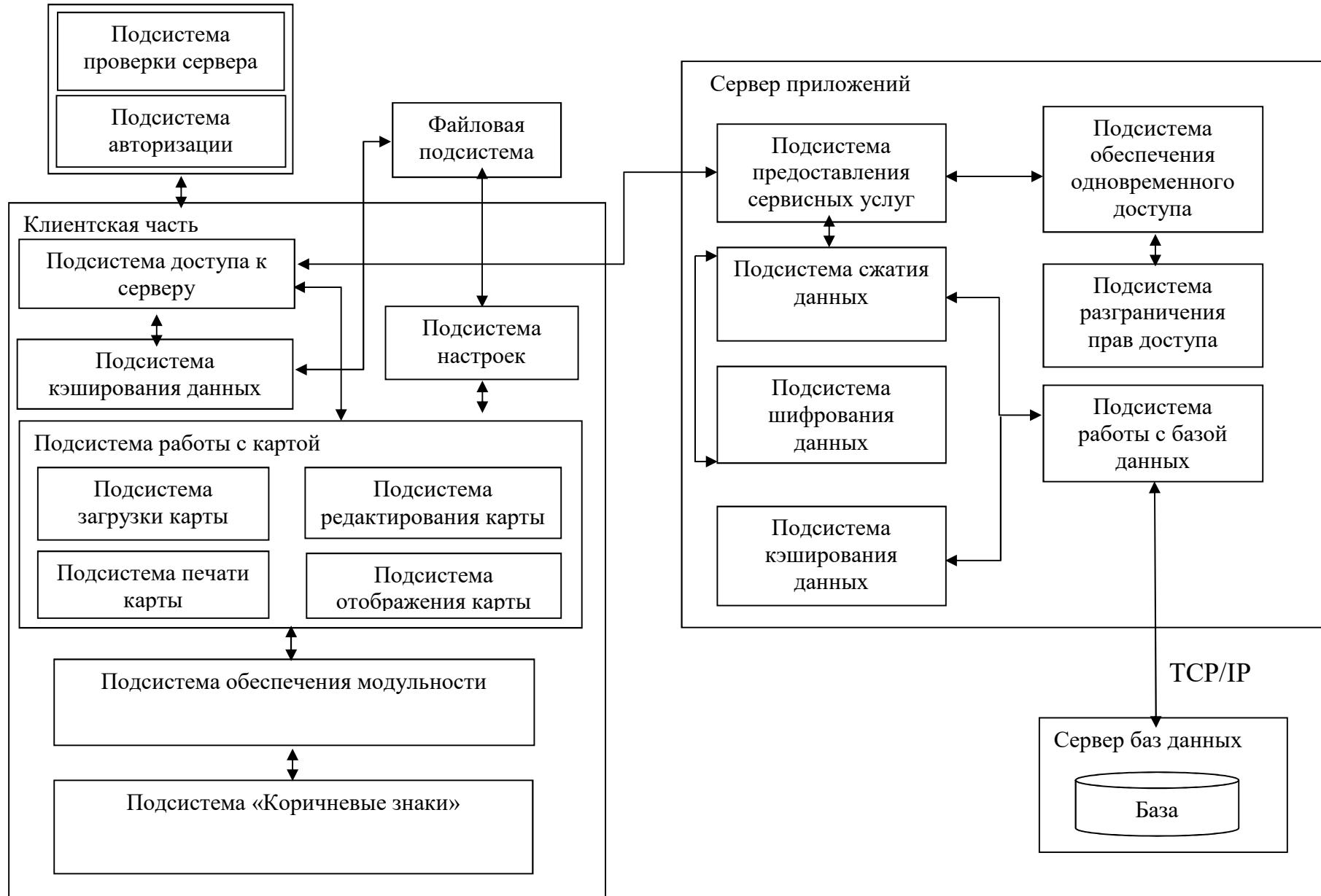


Рисунок 16 – Структурная схема ITSGIS

- подсистема работы с картой, которая включает в себя подсистемы:
 - 1 подсистема печати карты;
 - 2 подсистема загрузки карты, позволяющая загружать необходимую карту;
 - 3 подсистема редактирования карты, отвечающая за возможность внесения изменений на выбранной карте;
 - 4 подсистема отображения, выполняющая работу с картой, обеспечивающая преобразование координат между различными системами координат;
- подсистема обеспечения модульности – это механизм плагинов, подключающий дополнительные модули к ГИС;
- подсистема предоставления сервисных услуг «поднимает» и поддерживает в рабочем состоянии конечные точки доступа сервера;
- подсистема обеспечения одновременного доступа, например, проверяет, чтобы два пользователя не редактировали одну и ту же геометрию;
- подсистема сжатия данных;
- подсистема разграничения прав доступа;
- подсистема работы с базой данных;
- подсистема шифрования данных.

В состав плагина «Коричневые знаки» должны быть включены следующие подсистемы:

- подсистема создания коричневых знаков, которая отвечает за создание знака, включая все необходимые записи в базе данных;
- подсистема редактирования коричневых знаков которая отвечает за редактирование и удаление знака, включая все связанные с знаком записи в базе данных.

Сам плагин включается в состав клиентской и серверной частей. Структурная схема плагина представлена на рисунке 17.

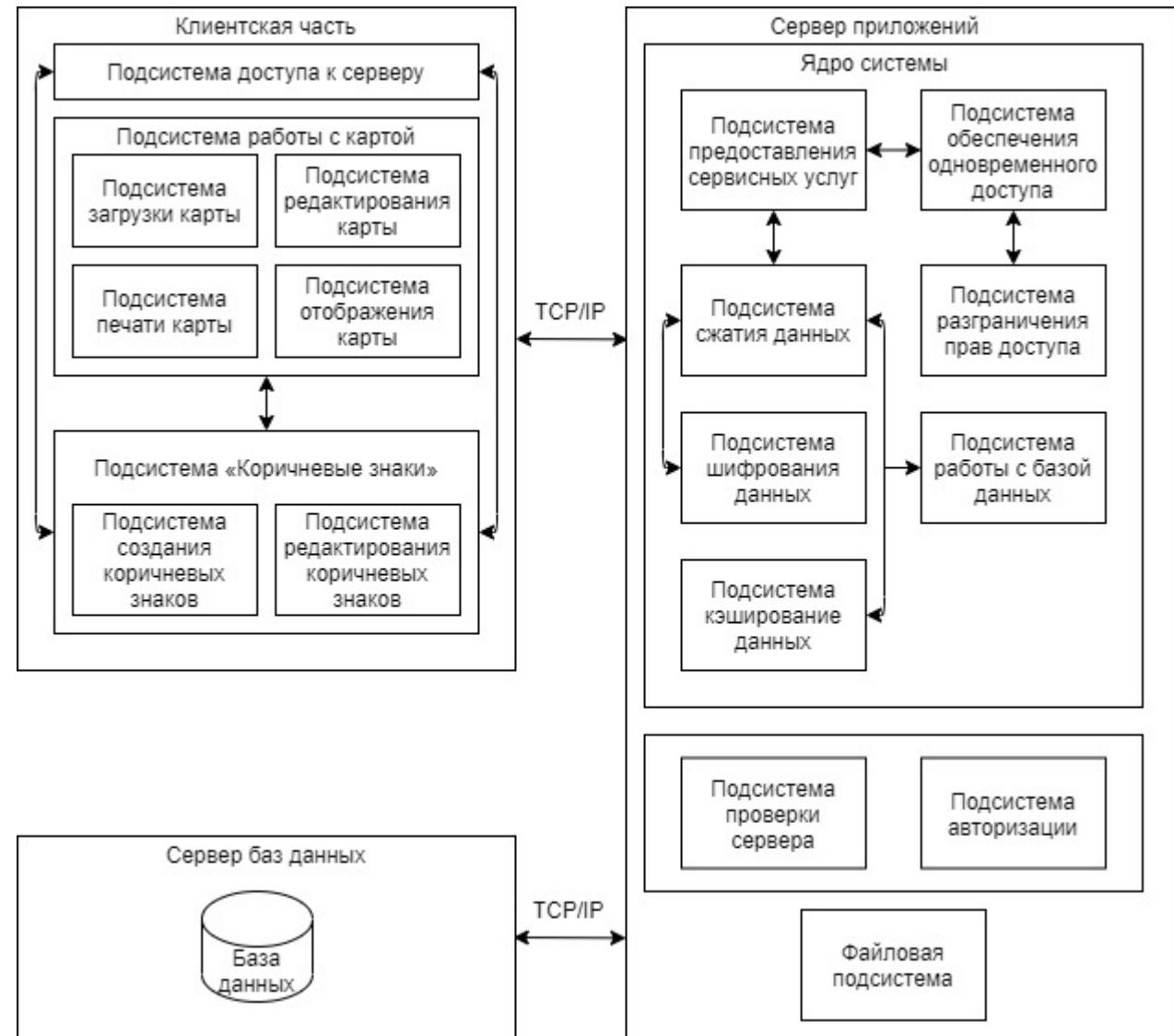


Рисунок 18 – Структурная схема плагина

2.3 Разработка прототипа интерфейса пользователя системы

Интерфейс пользователя является обязательной составляющей, программной системы. Недочёты в интерфейсе могут сильно испортить впечатления конечного пользователя о программе, поэтому нужно уделять особое внимание проектированию и разработке пользовательского интерфейса.

Пользовательский интерфейс – интерфейс, который позволяет передавать информацию между человеком-пользователем и аппаратными или программными компонентами компьютерной системы, или совокупность программных и аппаратных средств, позволяющих пользователю взаимодействовать с компьютерной системой [12].

В таблице 2 приведено краткое описание функционального описания панели управления плагином «Коричневые знаки».

Таблица 2 – Функциональное описание панели управления плагином.

Изображение	Всплывающая подсказка	Функционал
1	2	3
	Выделить знак	Выделяется на карте коричневый знак, «подсвечивая» красной окружностью, по нажатию на эту кнопку и затем нажатию на коричневый знак.
	Добавить знак	Открывается окно «Информация о знаке», в котором задаётся информация о создаваемом знаке. Для создания знака необходимо нажать на эту кнопку, затем нажать ЛКМ по точке на карте (указать местоположение).
	Удалить знак	Удаление выделенного знака.

Продолжение таблицы 2

1	2	3
	Переместить знак	Перемещение знака. Для перемещения знака необходимо нажать эту кнопку, затем зажав ЛКМ знак перемещать его.
	Редактировать знак	По нажатию открывается окно «Информация о знаке», в котором пользователь редактирует информацию о знаке.
	Информация о знаке	По нажатию открывается окно «Информация о знаке».
	Повернуть знак	По нажатию открывается диалоговое окно поворота знака на указанное количество градусов.
	Ведомость коричневых знаков	По нажатию открывается окно «Сводная ведомость коричневых знаков».
	Отмена	По нажатию отменяется предыдущее действие пользователя, связанное с коричневым знаком.
	Возврат	По нажатию возвращается отмененное действие пользователя, связанное с коричневым знаком
	Сохранить изменения	По нажатию сохраняются изменения, связанные с коричневыми знаками, действия, которые можно отменить «забываются» программой.
	Удалить изменения	По нажатию удаляются предыдущие действия пользователя, связанные с коричневыми знаками.

На рисунке 19 приведён прототип панели управления плагином. На панели расположены кнопки, описанные ранее в таблице 2. Панель доступна при выборе вкладки «Коричневые знаки».



Рисунок 19 – Прототип панели управления плагином

На рисунке 20 приведён прототип экранной формы просмотра информации о знаке.

A screenshot of a software window titled 'Информация о знаке'. At the top, there are three tabs: 'Основная информация' (Main information), 'Фото' (Photo), and 'Собственник' (Owner). Below the tabs is a large rectangular area containing a brown road sign with white text that reads 'МОСТ ЧЕРЕЗ РЕКУ ЯРЫНЬЯ 1.5'. Underneath the sign, there is a section titled 'Создать' (Create) with several input fields and dropdown menus. The fields include: 'Тип' (Type) set to 'Исторический' (Historical); 'Статус' (Status) set to 'Установлен' (Installed); 'Размеры (ВxH), мм' (Dimensions (WxH), mm) set to '2816x456'; 'Опора:' (Pylon:); 'Высота, см' (Height) set to '200'; 'Толщина, см' (Thickness) set to '5'; and 'Количество опор' (Number of pylons) set to '2'. At the bottom of the window are two buttons: 'OK' and 'Отмена' (Cancel).

Рисунок 20 – Прототип экранной формы просмотра информации о знаке

Здесь должна отображаться информация о знаке, доступная пользователю. Пользователь сможет изменить статус знака, высоту и толщину опоры, количество опор. По нажатию на кнопку «OK» изменения должны сохраниться в базу данных, по нажатию на «Отмена» – не будут

сохраняется, в обоих случаях форма должна быть закрыта. Для создания изображения знака необходимо нажать на гиперссылку «Создать», откроется форма «Редактор знака». На рисунке 21 приведён прототип экранной формы редактора знака.

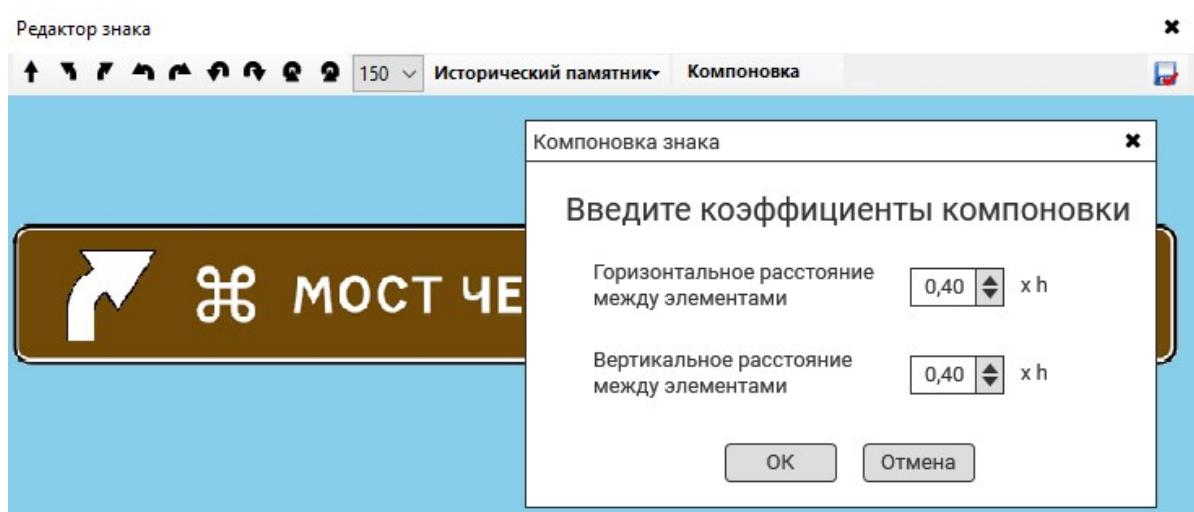


Рисунок 21 – Прототип экранной формы редактора знака

Размеры знака будут устанавливаться автоматически в соответствии с его содержимым. При выборе стрелки и нажатии на прямоугольник знака стрелка будет добавляться на знак, если стрелка уже добавлена на знак, то при выборе другой стрелки текущая стрелка заменится на выбранную.

Справа от стрелок будет расположен комбобокс с возможными реальными размерами букв на надписи знака, например, 150 мм, отображение букв надписей в редакторе знака должно быть пропорционально размеру букв надписей в реальном знаке.

Справа от комбобокса будет расположена другая комбобокс с возможными типами коричневым знаков, например, «Исторический знак». При выборе значения комбобокса на знак будет добавляться пиктограмма, соответствующая выбранному типу коричневого знака.

По нажатию на кнопку «Компоновка» будет открываться диалоговое окно, в котором задаются горизонтальные и вертикальные расстояния между элементами знака (между стрелками, пиктограммами и надписями).

Для добавления надписи на знак необходимо будет дважды нажать левую кнопку мыши (ЛКМ) по прямоугольнику знака, затем ввести при

помощи клавиатуры надпись. Возможно создание второй строки знака (второй надписи), для этого будет необходимо нажать клавишу Enter после ввода надписи. Для второй надписи стрелка и пиктограмма будут добавляться аналогично добавлению для первой надписи.

В правом углу будет расположена иконка в виде дискеты, по нажатию на которую созданное изображение знака должно сохраняться, и форма должна будет закрыться. По нажатию на **×** созданный знак не будет сохраняться, форма закроется.

Экранная форма «Коричневые знаки» будет иметь три вкладки: «Основная информация», «Фото», «Собственник».

На рисунке 22 представлена вкладка «Фото» прототипа экранной формы, представленного на рисунке 20.

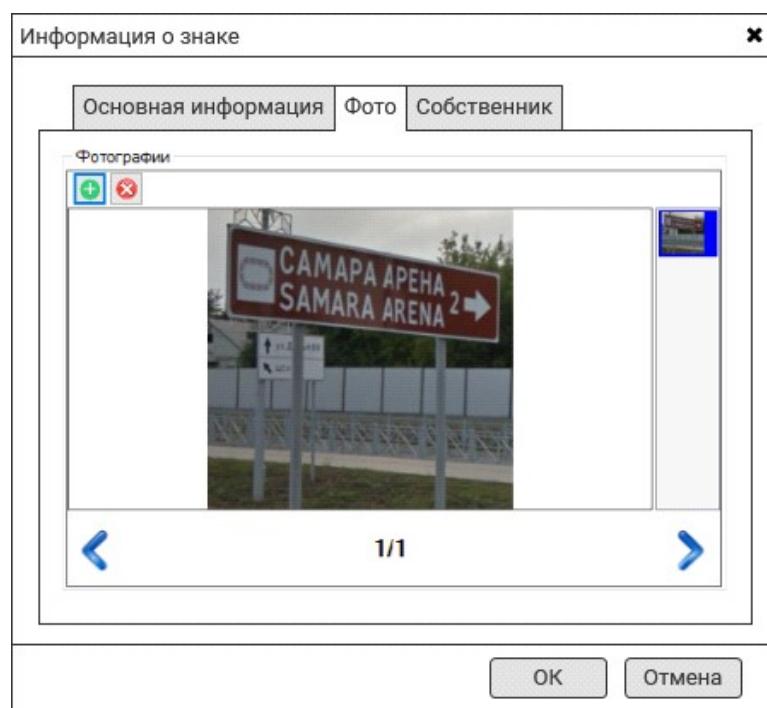


Рисунок 22 – Прототип экранной формы просмотра информации о знаке.

Вкладка «Фото»

По нажатию на **+** откроется окно операционной системы выбора изображения, при выборе изображения и нажатия кнопки «ОК» изображение будет доступно во вкладке «Фото». По нажатию на **×** выбранное фото удалится из вкладки.

Все доступные изображения будут отображены в виде значков на правой вертикальной панели. На этой панели можно будет выбрать фото для просмотра, выбранное фото отобразится на центральной панели вкладки. Возможно переключение между изображениями с помощью синих стрелок

По нажатию на кнопку «OK» вкладки «Фото» изменения сохранятся (например, добавления и удаления фото), по нажатию на «Отмена» – не сохраняются.

На рисунке 23 представлен прототип вкладки «Собственник» экранной формы «Информация о знаке».

Здесь будет отображаться доступная пользователю информация о собственнике, возможно редактирование этой информации.

The screenshot shows a window titled 'Информация о знаке' (Information about a sign) with a close button 'x'. Below the title bar is a horizontal menu bar containing three tabs: 'Основная информация' (Main information), 'Фото' (Photo), and 'Собственник' (Owner). The 'Основная информация' tab is currently selected. The main content area displays four data entries in a grid format:

Название	Новгородский музей-заповедник
Почтовый адрес	Новгородский детинец
Телефон	8-921-730-93-92
Веб-сайт	https://novgorodmuseum.ru/

At the bottom of the window are two buttons: 'OK' and 'Отмена' (Cancel).

Рисунок 23 – Прототип экранной формы просмотра информации о знаке.

Вкладка «Собственник»

На рисунке 24 представлен прототип экранной формы «Сводная ведомость знаков».

По нажатию на кнопку «Найти» сформируется сводная ведомость коричневых знаков, она должна отображаться в виде таблицы. По нажатию на кнопку «Скачать» сформированная ведомость загрузится на запоминающее устройство пользователя.

Сводная ведомость знаков			
	Адрес	Номер по ГОСТ	Наименование
📍 ↗	ул. Ново-Садовая	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	ул. Ново-Садовая	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	ул. Аминева	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	ул. Аминева	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	ул. Солнечная	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	НОВО-САДОВАЯ УЛИЦА	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	КИРОВА ПРОСПЕКТ	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ
📍 ↗	ул. Ново-Садовая	6.10	УКАЗАТЕЛЬ НАПРАВЛЕНИЙ

Рисунок 24 – Прототип экранной формы «Сводная ведомость знаков»

На Рисунке приведена навигационная модель разрабатываемого плагина.

Из панели управления плагином пользователь сможет вызвать две формы: «Информация о знаке» и «Сводная ведомость знаков».

В соответствии с нажатой кнопки на панели управления форма «Информация о знаке» должна будет вызваться в некотором виде: для создания знака, для редактирования информации о знаке, для просмотра информации о знаке.

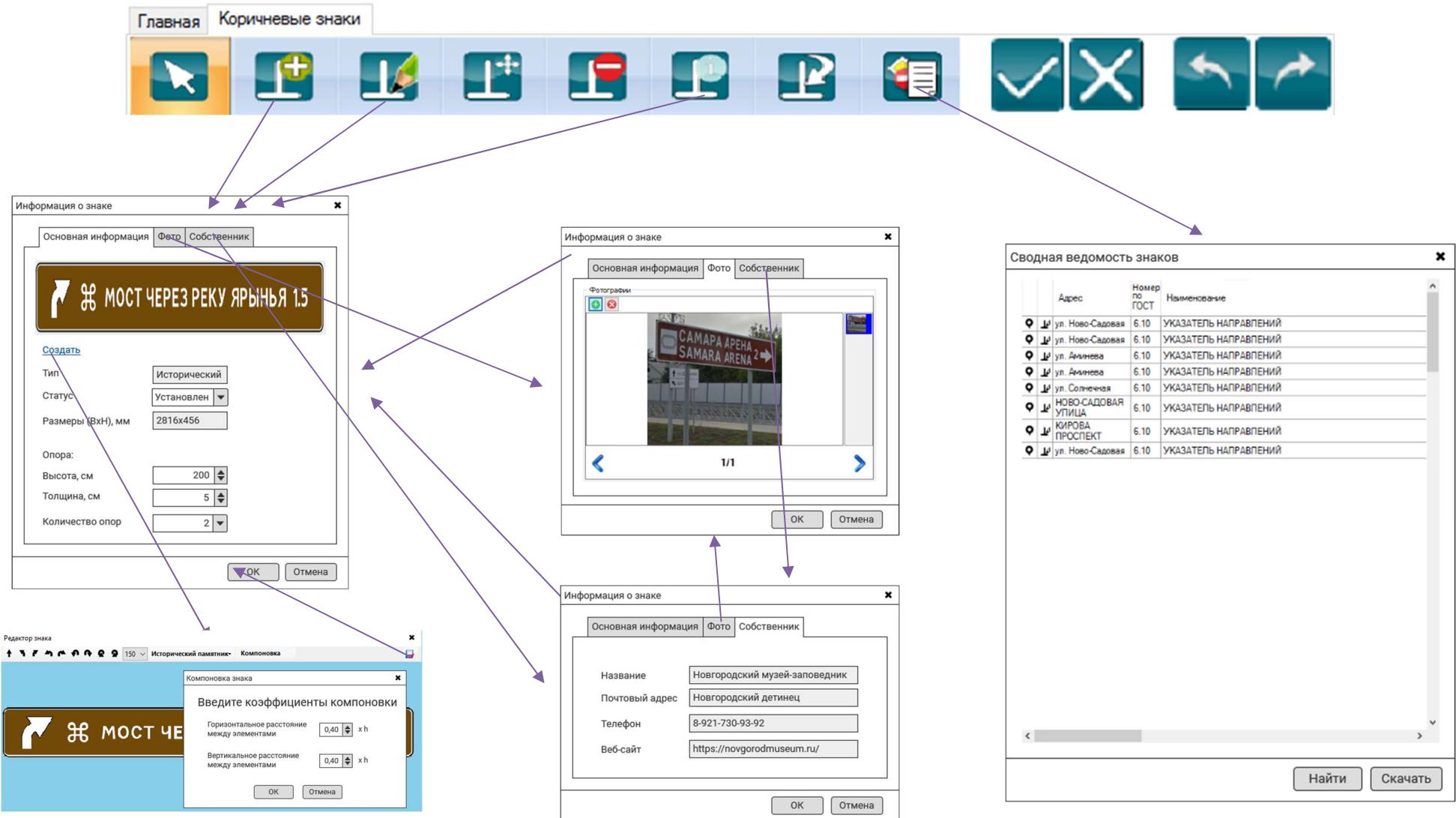


Рисунок 24 ┌ Навигационная модель плагина «Коричневые знаки»

2.4 Разработка информационно-логического проекта системы

2.4.1 Язык UML

UML (англ. Unified Modeling Language — унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения [13].

UML предоставляет разработчикам программного обеспечения единство в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение и прочие) и позволяет разработчикам сконцентрироваться на проектировании и архитектуре программного обеспечения.

Для специфирования системы, разрабатываемой в данной работе, и её документирования используется унифицированный язык моделирования UML.

2.4.2 Диаграмма вариантов использования

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. На ней изображаются отношения между акторами и вариантами использования [13].

На рисунке 25 приведена диаграмма вариантов использования (пользователя). В системе существуют две роли – администратор и пользователь. Авторизованный пользователь может просматривать сводную ведомость коричневых знаков, информацию о выбранном геообъекте «Коричневый знак», неавторизованный пользователь может только зарегистрироваться в системе. Работать с коричневым знаком (редактировать его) может только пользователь, авторизованный как администратор.

На рисунке 26 приведена диаграмма вариантов использования, уточняющая вариант использования «Работать с картой». Администратор может создать, редактировать, удалить, переместить и повернуть коричневый

знак, а также сохранять и отменять изменения геообъекта «Коричневый знак».

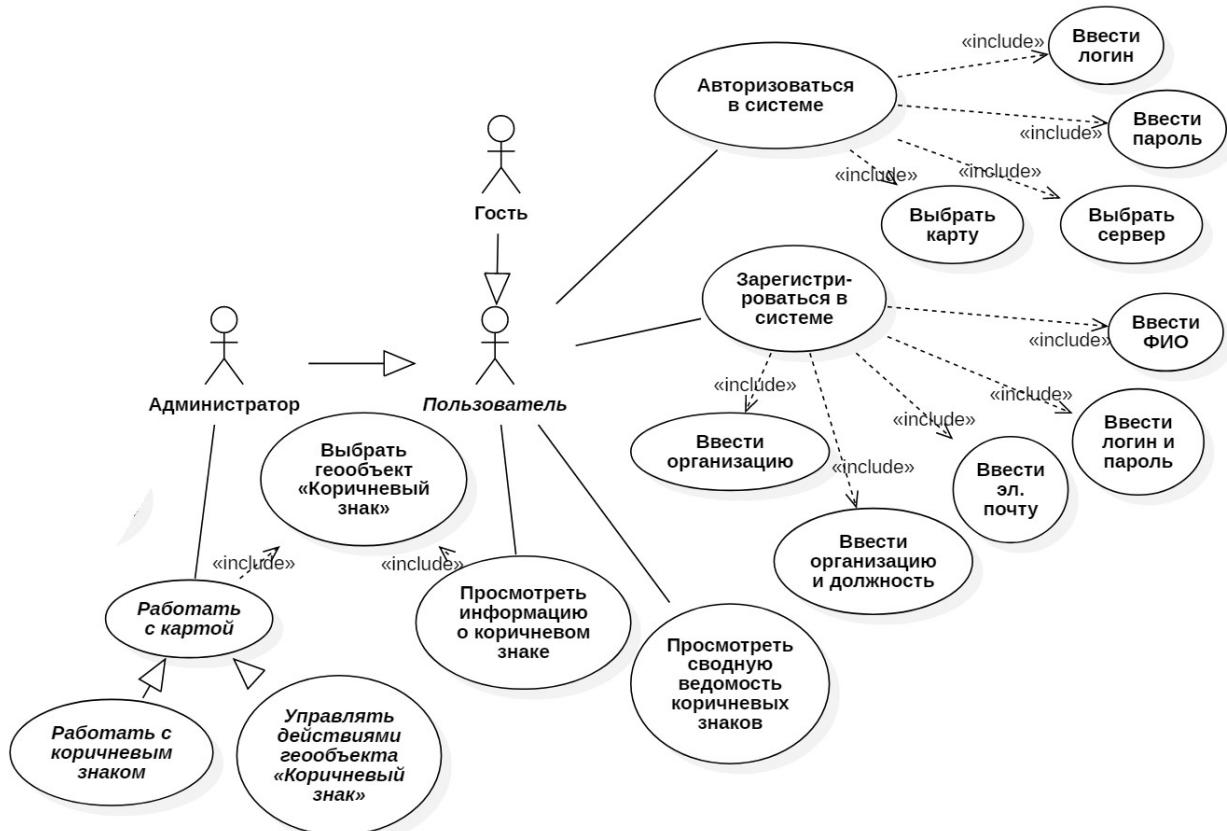


Рисунок 25 – Диаграмма вариантов использования



Рисунок 26 – Диаграмма вариантов использования, уточняющая вариант «Работать с картой»

На рисунке 27 приведена диаграмма вариантов использования, уточняющая вариант использования «Работать с информацией о коричневом знаке». При создании коричневого знака, администратор должен задать следующие характеристики знака:

- надписи на знаке (при необходимости высоту прописных букв надписей);
 - размеры знака;
 - выбрать тип коричневого знака (например, «Музей»);
 - выбрать статус знака (например, «Установлен»);
 - выбрать типы опоры (например, «Две опоры»);
 - задать собственника знака;
 - задать направление стрелки на знаке;
 - при необходимости добавить или удалить фото знака.

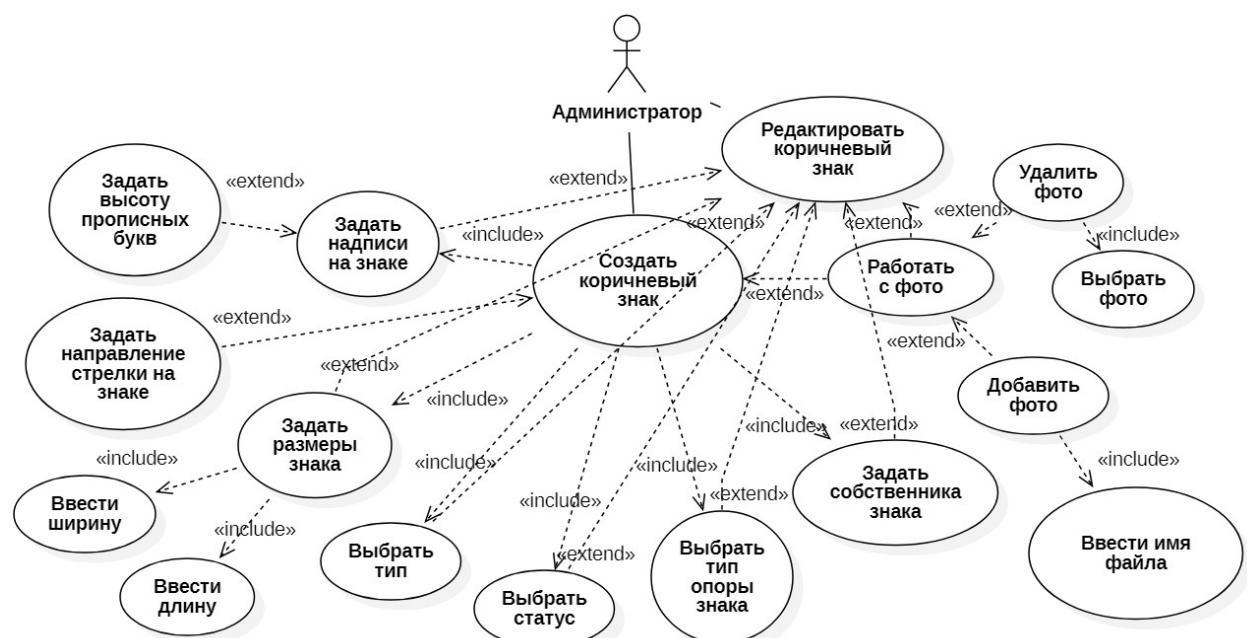


Рисунок 27 – Диаграмма вариантов использования, уточняющая вариант «Работать с информацией о коричневом знаке»

2.4.3 Сценарии

Сценарий – это конкретная последовательность действий, иллюстрирующих поведение. Сценарии по отношению к вариантам

использования – то же самое, что экземпляры по отношению к классам, поскольку сценарий – это в основном один экземпляр варианта использования [13].

В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования.

Рассмотрим вариант использования «Задать надписи на знаке».

Краткое описание: Даёт администратору возможность задать надписи на коричневом знаке.

Актант: администратор.

Предусловия: компьютер пользователя включён, система ITSGIS запущена, администратор авторизовался, выбрал вкладку «Коричневые знаки», выбрал точку на карте, нажал на кнопку «Добавить знак» на панели управления плагином «Коричневые знаки», открыл форму «Редактор знака», перейдя по ссылки «Создать» в форме «Информация о знаке».

Основной поток событий:

1 На экране отображается форма редактор «Редактор знака» (см. рисунок 21). В центре расположена заготовка знака в виде коричневого прямоугольника. На верхней панели формы расположены блок кнопок со стрелками, выпадающее меню с размерами букв надписей, выпадающее меню с типами коричневого знака и кнопка «Компоновка».

2 Администратор дважды щёлкает курсором по коричневому прямоугольнику.

3 Администратор вводит с клавиатуры надпись на знаке.

4 Администратор нажимает на кнопку  . Система сохраняет созданное изображение знака и закрывает экранную форму «Редактор знака». Вариант использования завершается успешно.

A1: Администратор нажимает клавишу Enter и вводит с клавиатуры следующую надпись на знаке.

A2: Администратор устанавливает курсор ввода на надписи, затем нажимает на любую иконку стрелки.

A3: Администратор выбирает высоту прописных букв в соответствующем комбобоксе.

A4: Администратор выбирает тип коричневого знака в соответствующем комбобоксе.

A5: Администратор нажимает на кнопку «Компоновка».

A6: Администратор нажимает на *****.

Альтернативы:

A1: Администратор нажимает клавишу Enter и вводит с клавиатуры следующую надпись на знаке.

A1.1: Действие A1 продолжается до тех пор пока администратор не введёт все необходимые ему надписи.

A1.2: Осуществляется переход к пункту 3 основного потока событий.

A2: Администратор устанавливает курсор ввода на надпись, затем нажимает на любую иконку стрелки.

A2.1: Система добавляет пиктограмму стрелки перед надписью, либо перед пиктограммой, если она уже добавлена перед надписью.

A2.2: Действие A2 продолжается до тех пор, пока администратор не добавит все необходимые ему стрелки для нужных надписей.

A2.3: Осуществляется переход к пункту 3 основного потока событий.

A3: Администратор выбирает высоту прописных букв в соответствующем комбобоксе.

A3.1: Осуществляется переход к пункту 3 основного потока событий.

A4: Администратор выбирает тип коричневого знака в соответствующем комбобоксе.

A4.1: Система добавляет для выбранной надписи пиктограмму, соответствующую выбранному типу, между надписью и стрелкой.

 A4.1.A1: Система добавляет пиктограмму перед надписью, если стрелка не добавлена.

 A4.1.A2: Система не добавляет пиктограмму перед надписью, если выбран тип «По умолчанию» (тип без пиктограммы).

A4.2: Осуществляется переход к пункту 3 основного потока событий.

A5: Администратор нажимает на кнопку «Компоновка».

 A5.1: Система открывает диалоговое окно «Компоновка знака».

 A5.2: Администратор вводит горизонтальное и вертикальное расстояние между элементами.

 A5.2.A1: Если одно из введённых расстояний меньше 0.4, то система заменяет на 0.4.

 A5.2.A2: Если одно из введённых расстояний больше 0.8, то система заменяет введённое значение на 0.8.

 A5.2.A3: Администратор вводит горизонтальное и вертикальное расстояние между элементами при помощи стрелочек «вверх» и «вниз».

 A5.2.A3.1: Система в соответствии с нажатой стрелочкой увеличивает или уменьшает значение на 0.1.

 A5.2.A3.1.A1: Если одно из введённых расстояний меньше 0.4, то система заменяет введённое значение на 0.4.

 A5.2.A3.1.A2: Если одно из введённых расстояний больше 0.8, то система заменяет введённое значение на 0.8.

 A5.3: Администратор нажимает на кнопку «OK».

 A5.3.1: Система сохраняет введённые расстояния.

A5.3.2: Система закрывает диалоговое окно.

A5.3.3: Система изменяет расстояния между элементами знака в соответствии с ранее введёнными расстояниями.

A5.3.3.A1: Администратор нажимает на кнопку «Отмена». Система закрывает диалоговое окно, расстояния между элементами знака не изменяются.

A5.3.4: Осуществляется переход к пункту 3 основного потока событий.

А6: Администратор нажимает на *****.

A6.1: Система не сохраняет созданное изображение знака, закрывает экранную форму «Редактор знака».

A6.2: Вариант использования завершается.

Постусловия: если изменения сохранены, то отображается форма «Информация о знаке» с изображённым на ней созданным коричневым знаком, иначе изображение коричневого знака отсутствует.

2.4.4 Диаграмма классов

Диаграммы классов – диаграммы, изображающие набор классов, интерфейсов и коопераций, а также их связи. Диаграммы классов используются для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения [13].

В таблице 3 приведено описание классов.

Таблица 3 – Описание классов системы

Название класса	Назначение
1	2
Контроллер коричневых знаков	Основной класс, описывает функции плагина. Содержит поля для ведения истории изменений и сохранённых изменений для возможности их отмены.

Продолжение таблицы 3

1	2
Коричневый знак	Класс, описывает объект коричневого знака. Содержит поля, описывающие основные характеристики геообъекта.
Опора	Класс, описывает объект опоры, поле такого типа имеется у объекта коричневого знака. Содержит поля, описывающие основные характеристики опоры.
Собственник	Класс, описывает объект собственника, поле такого типа имеется у объекта коричневого знака. Содержит поля, описывающие основные характеристики собственника.
Перечисление статусов знаков	Перечисление возможных статусов знаков.
Перечисление высот прописных букв	Перечисление возможных высот прописных букв коричневого знака.
Перечисление направлений стрелки	Перечисление направлений стрелки на коричневом знаке.
Перечисление типов коричневых знаков	Перечисление возможных типов коричневого знака.
Перечисление количества опор	Перечисление возможного количества опор знака.

На рисунке 28 приведена диаграмма классов системы (этап проектирования).

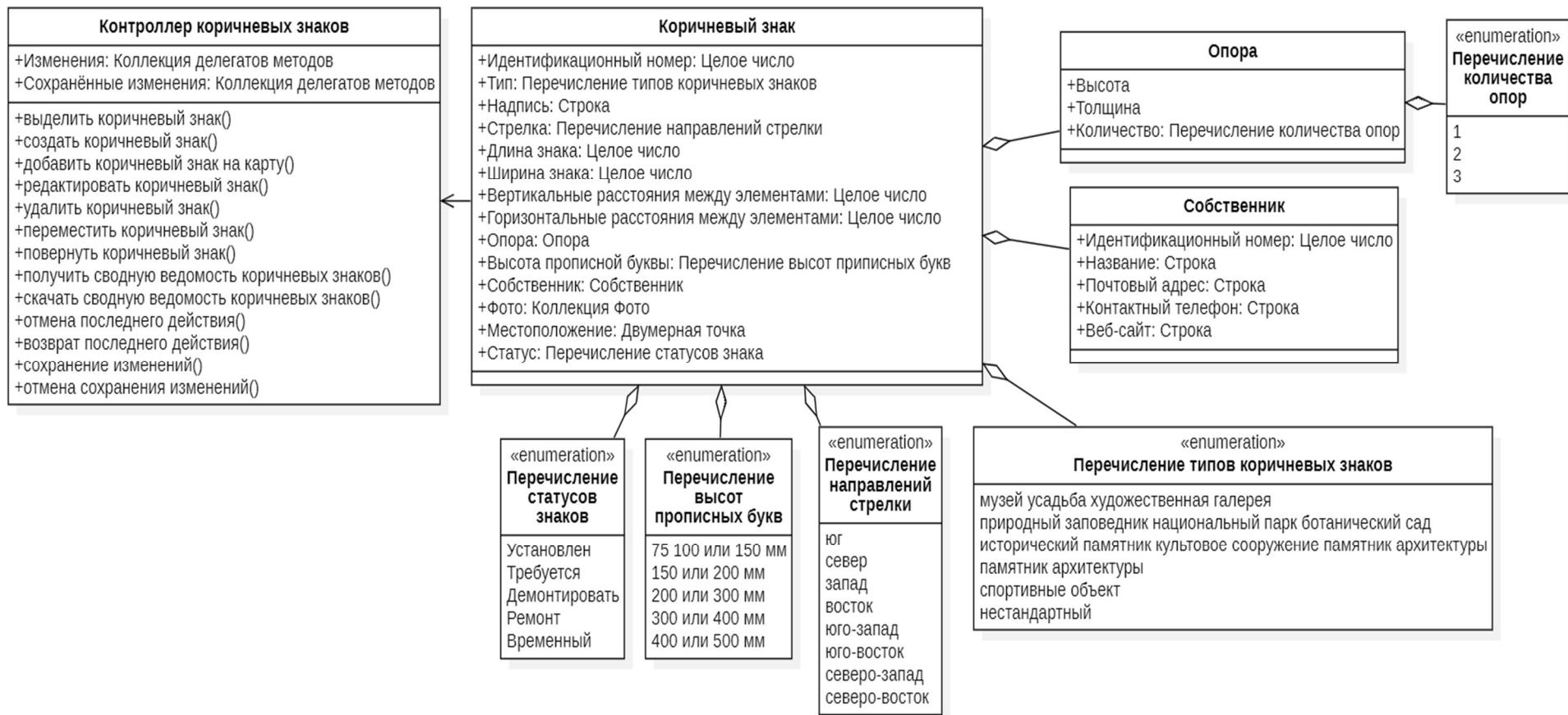


Рисунок 28 – Диаграмма классов плагина «Коричневые знаки»

2.4.5 Диаграмма состояний

Диаграмма состояний – это диаграмма UML, предназначенная для моделирования динамических аспектов поведения систем. Диаграмма состояний показывает конечный автомат [13].

Диаграммы состояния применяются для моделирования динамических аспектов поведения систем (в большинстве случаев это моделирование поведения реактивных объектов). Реактивным называется такой объект, поведение которого лучше всего характеризуется его реакцией на события, получаемые им извне его контекста. Реактивный объект имеет чёткое время жизни, и его текущее поведение зависит от прошлого.

Общая диаграмма состояний представлена на рисунке 29. Диаграмма описывает поведение плагина до авторизации пользователя, указаны возможность авторизации с правами администратора, описаны состояния начала и завершения работы с приложением ITSGIS со встроенным в него плагином «Коричневые знаки». Пользователи, в независимости от имеющихся у них прав, могут просматривать информацию о выбранном геообъекте коричневого знака и могут получить сводную ведомость обо всех коричневых знаках.

Декомпозиция состояния «Приложение работает в режиме администратора» представлена на рисунке 30. Декомпозиция описывает поведение приложения в случае авторизации пользователя, имеющего права администратора. Указаны возможности добавления, перемещения, поворота, редактирования и удаления коричневых знаков, а также отмены и возвращения последнего действия, сохранения и отмены изменений.

2.4.6 Диаграмма деятельности

Диаграмма деятельности (активности – Activity diagram) применяется для моделирования динамических аспектов поведения системы [13]. Это частный случай диаграмм состояний, они позволяют реализовать

особенности процедурного и синхронного управления, обусловленного завершением внутренних действий и деятельности.

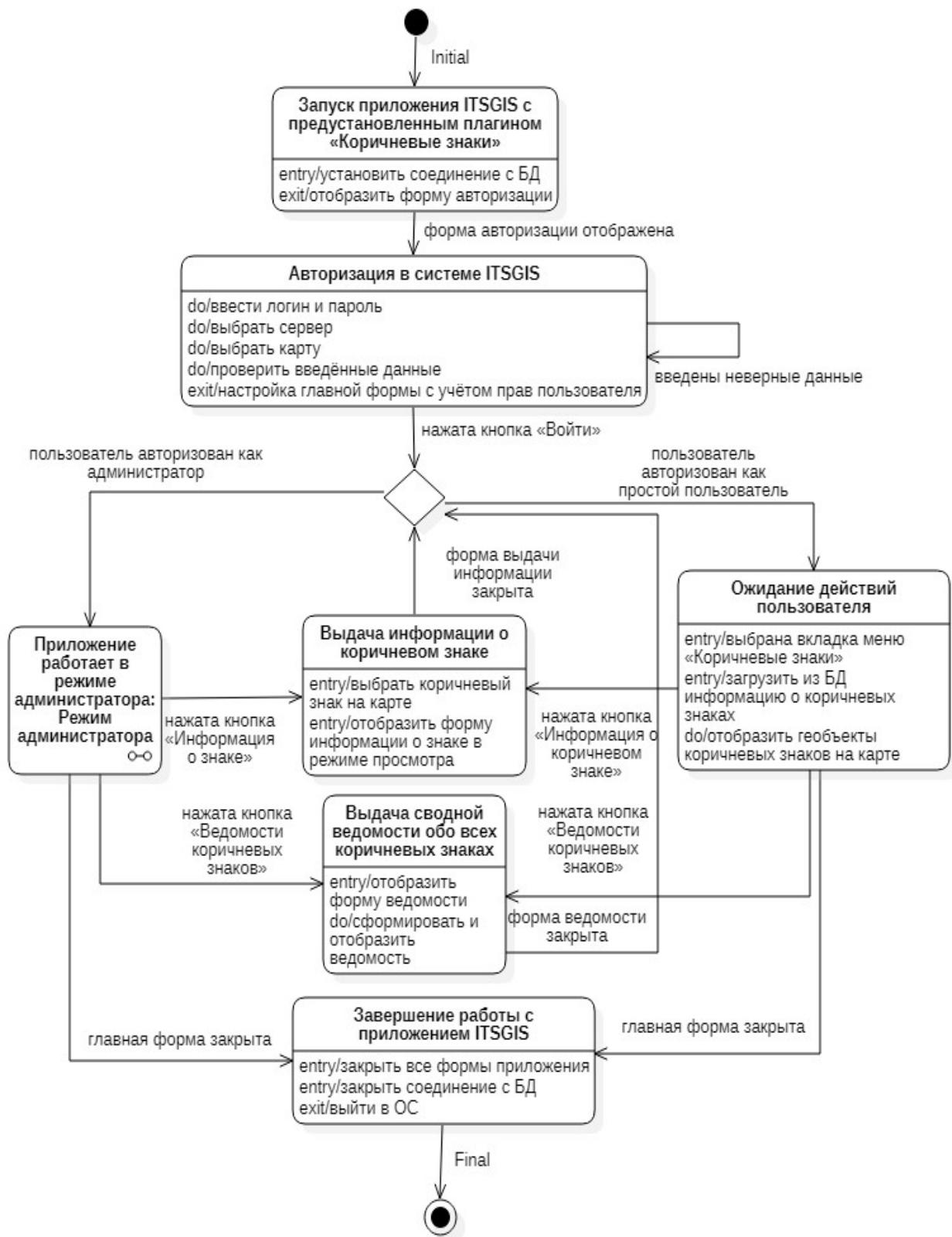


Рисунок 29 – Общая диаграмма состояний системы

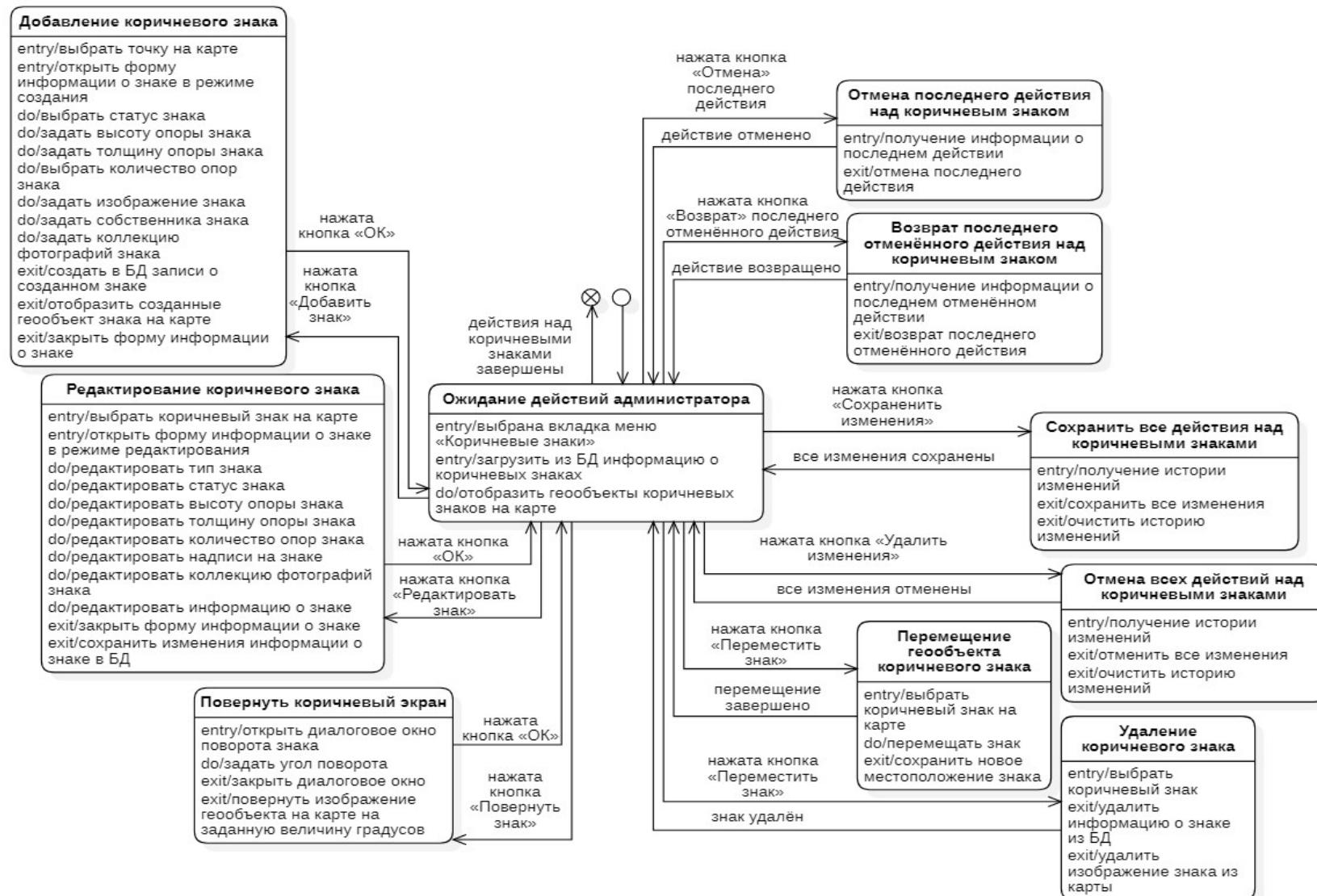


Рисунок 30 – Декомпозиция диаграммы состояний «Работа в режиме администратора»

На рисунке 31 приведена диаграмма деятельности для варианта использования «Добавить знак».

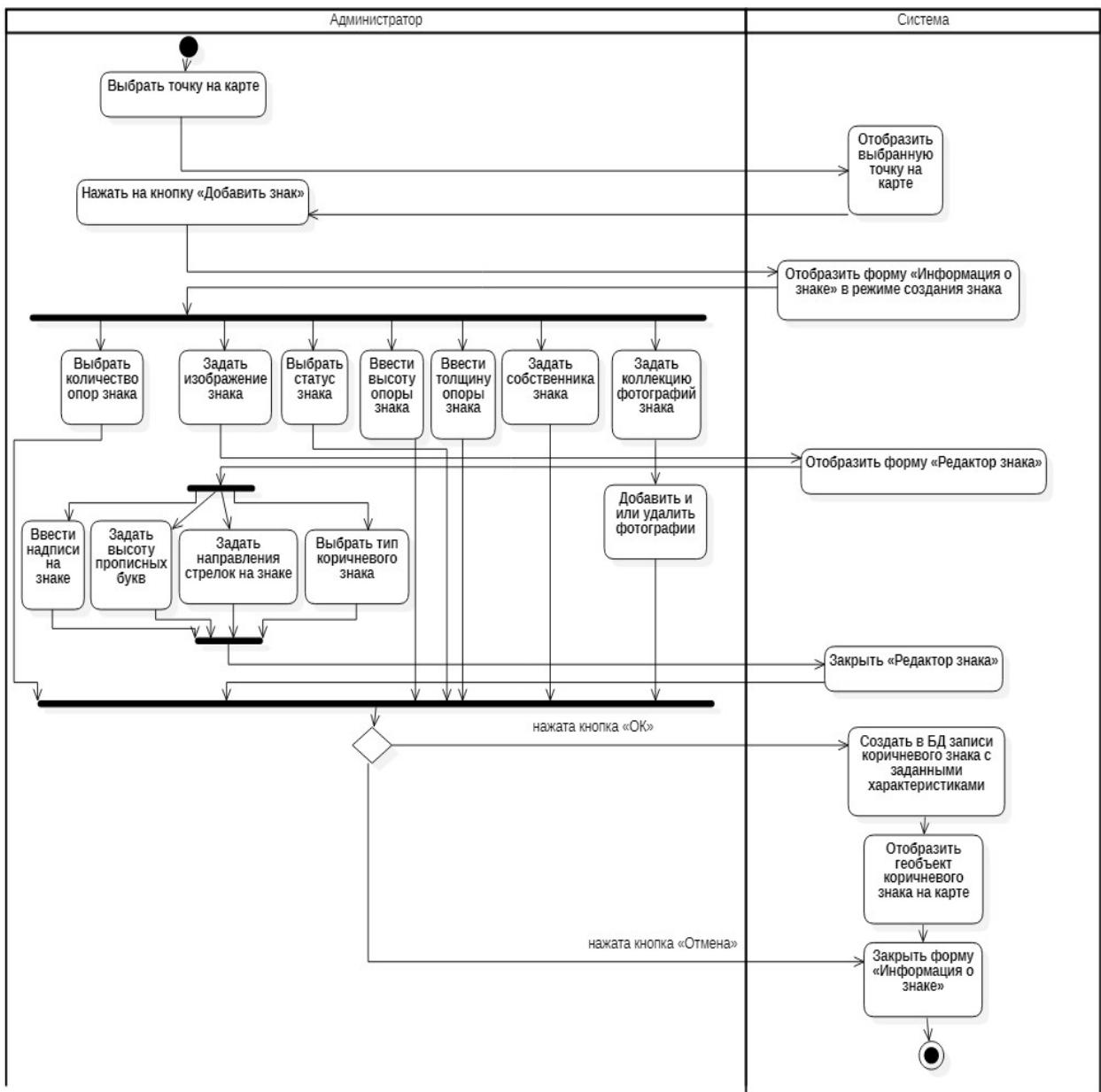


Рисунок 31 – Диаграмма деятельности добавления знака

На диаграмме изображено следующее. Администратор выбирает точку на карте, система отображает эту точку на карте. Затем пользователь нажимает на кнопку «Добавить знак», система отображает форму «Информация о знаке», в этой форме пользователь задаёт характеристики знака, а также вызывает из неё «Редактор знака» для создания изображения знака. В случае нажатия администратором кнопки «OK» система создаёт записи в базе данных, связанные с коричневым знаком с заданными

администратором характеристиками, и отображает созданный коричневый знак в виде геообъекта на карте. В случае, когда пользователь нажал на кнопку «Отмена», коричневый знак не создаётся.

2.4.7 Диаграмма последовательности

Диаграмма последовательности – своеобразный временной график «жизни» всей совокупности объектов, связанных между собой для реализации варианта использования программной системы, достижения бизнес-цели или выполнения какой-либо задачи. позволяет визуализировать временные отношения между передаваемыми сообщениями [13]. На диаграмме, вертикально проходят линии жизни объектов, для представления коммуникации между которыми используются разного рода сообщения.

На рисунке 32 приведена диаграмма последовательности системы для варианта использования «Задать надписи на знаке» – администратор задаёт надписи на коричневом знаке.

2.5 Логическая модель данных

Логическая информационная модель – модель данных, в которой учитывается способ логического хранения данных в памяти ЭВМ [14]. При построении модели базы данных (БД) используются следующие понятия.

Сущность – объект предметной области, который можно выделить исходя из сути предметной области, для которой разрабатывается база данных. Сущность состоит из множества своих экземпляров. Каждая сущность обладает свойствами – атрибутами.

Атрибут – определённое свойство сущности. В общем случае набор атрибутов для каждой сущности унаследован [14].

Атрибут или набор атрибутов, используемый для идентификации экземпляра сущности, называется ключом сущности. Каждый экземпляр сущности однозначно определяется ключом.

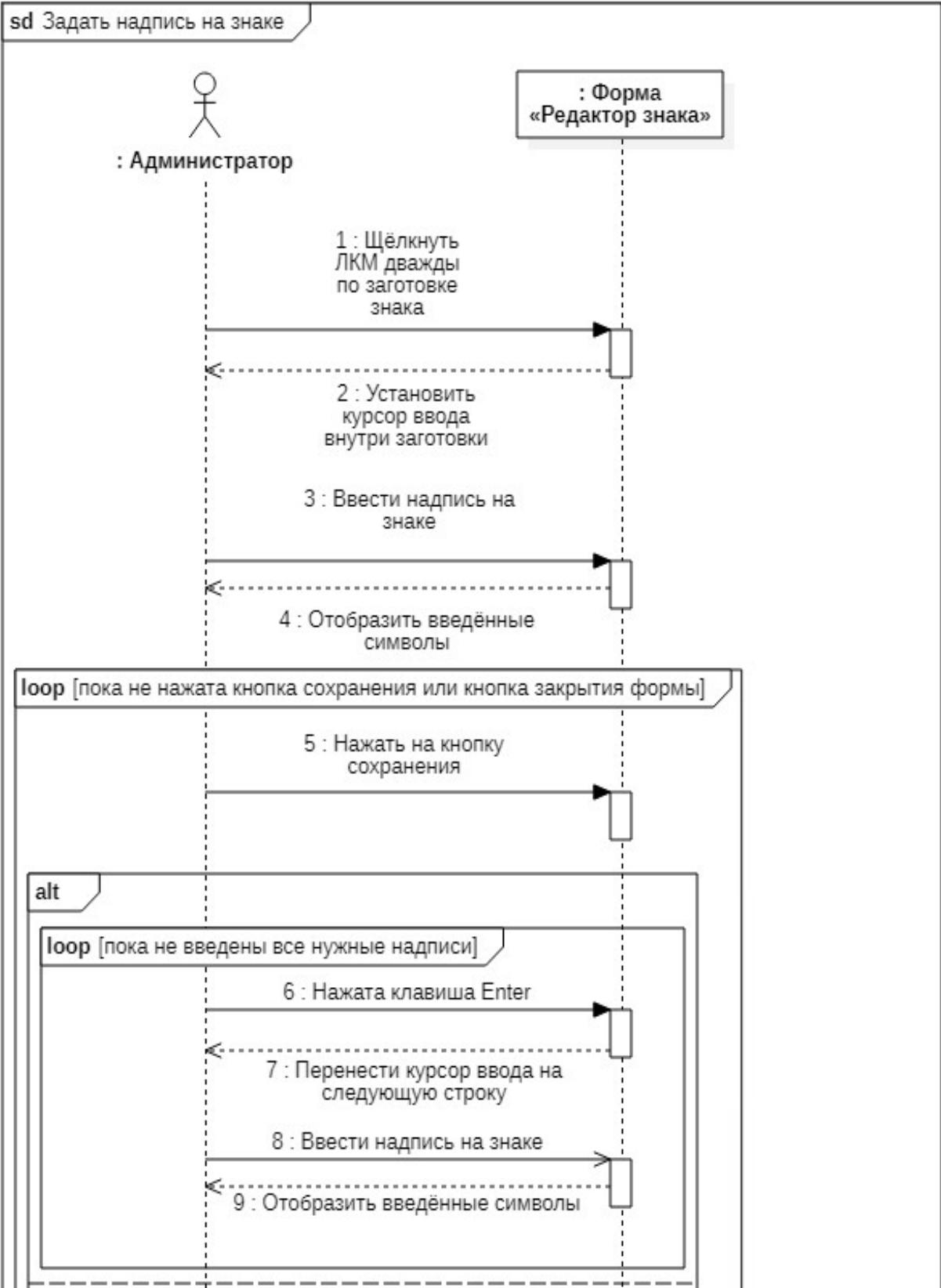


Рисунок 32 – Диаграмма последовательности задания надписи на знаке
(начало)

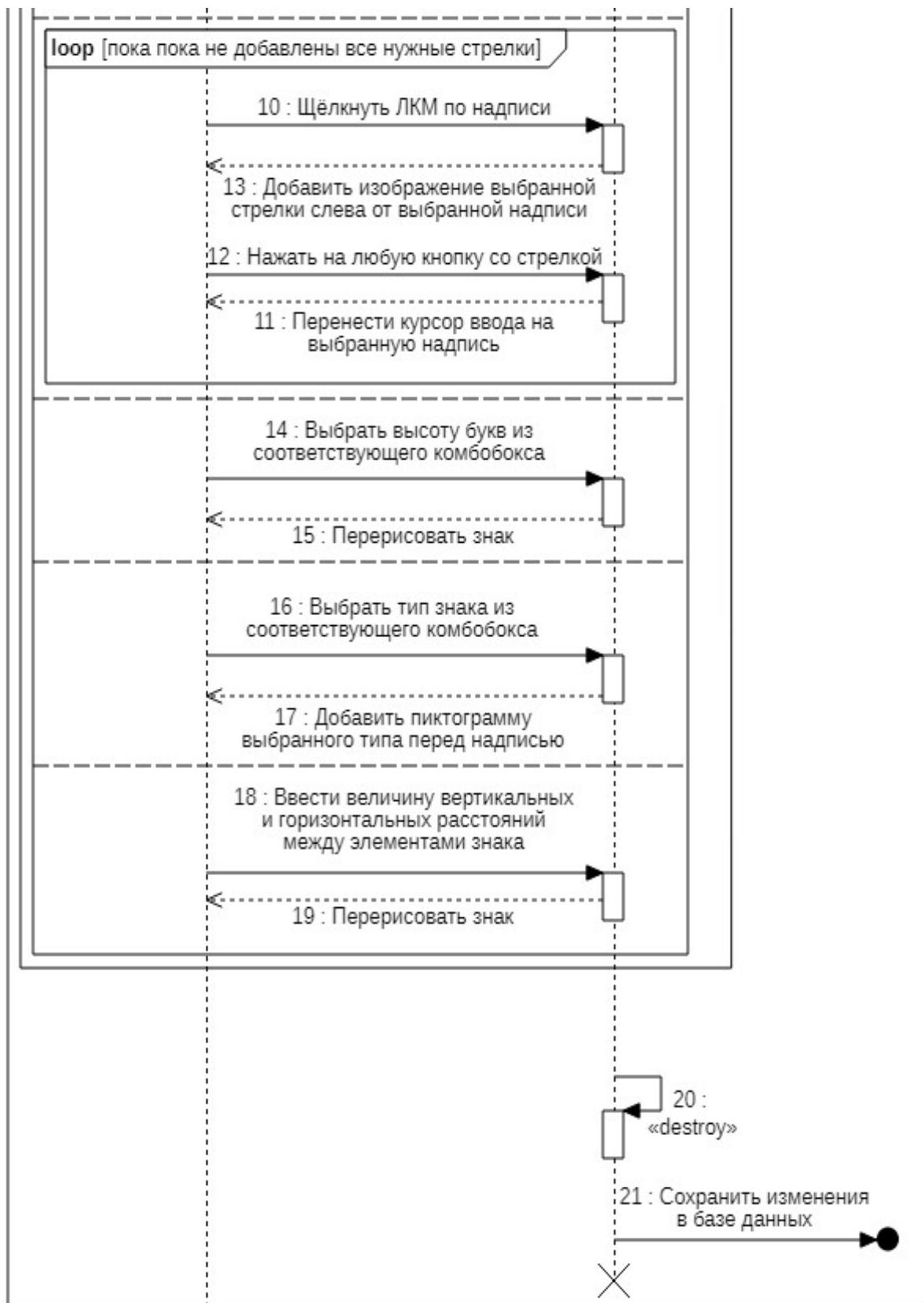


Рисунок 32 – Диаграмма последовательности задания надписи на знаке
(окончание)

На рисунке 32 представлена логическая модель базы данных плагина «Коричневые знаки». Описание входящих в нее сущностей содержится в таблице 3.

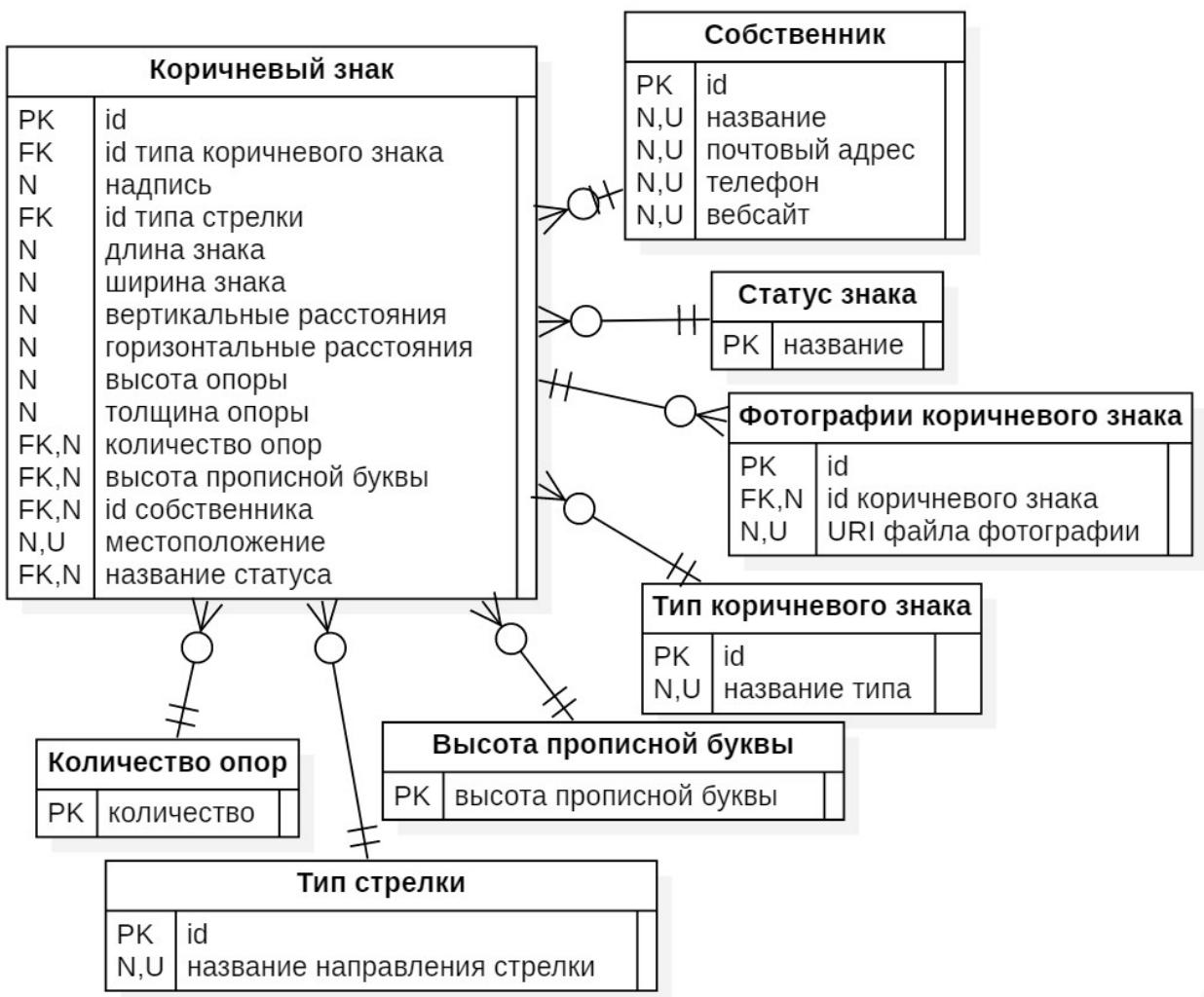


Рисунок 32 – Логическая модель базы данных плагина «Коричневые знаки»
Таблица 3 – Описание сущностей базы данных плагина «Коричневые знаки»

Название	Назначение
1	2
Коричневый знак	Основная сущность. Содержит характеристики коричневого знака.
Собственник	Сущность описывает характеристики собственника. Ссылка на собственника (внешний ключ) имеется в сущности коричневого знака – собственник владеет коричневым знаком.

Продолжение таблицы 3

1	2
Фотографии коричневого знака	Сущность хранит URI каждой фотографии для каждого коричневого знака – коричневый знак может иметь несколько фотографий.
Количество опор	Сущность представляет перечисление возможного количества опор у знака. Каждый знак имеет опору – каждая сущность коричневого знака имеет ссылку на количество используемых опор. В качестве первичного ключа выбрано количество опор (число).
Тип стрелки	Сущность представляет возможные направления стрелки. Надпись на коричневом знаке может иметь направление, поэтому каждая сущность коричневого знака имеет ссылку сущность типа стрелки.
Высота прописной буквы	Сущность представляет возможную высоту буквы коричневого знака. В качестве первичного ключа выбрана высота буквы (число).
Тип коричневого знака	Сущность представляет тип коричневого знака. Каждый коричневый знак представляет какой-то тип, например, музей, поэтому сущность коричневого знака имеет ссылку на сущность типа коричневого знака.
Статус знака	Сущность представляет возможные статусы коричневого знака, например, «установлен». В качестве первичного ключа выбрано название статуса.

2.6 Описание комплекса программных средств

Программное средство – это объект, состоящий из программ, процедур, правил, а также сопутствующих им документации и данных, относящихся к функционированию системы обработки информации [15].

Программное средство представляет собой конкретную информацию, объективно существующую как совокупность всех значимых с точки зрения ее представления свойств каждого из материальных объектов, содержащих в фиксированном виде эту информацию.

2.6.1 Описания языка программирования и среды разработки

Язык программирования – формальный язык, предназначенный для представления программ [16]. В качестве языка написания основной логики системы ITSGIS используется C#.

C# использует кроссплатформенную среду исполнения .NET Framework и программный интерфейс для разработки графических приложений Windows Forms.

С применением C# и .NET Framework можно создавать динамические веб-страницы, графические интерфейсы Windows Presentation Foundation, веб-службы XML, компоненты для распределенных приложений, компоненты для доступа к базам данных, классические настольные приложения Windows и даже клиентские приложения, обладающие возможностями для работы в оперативном и автономном режимах

Кроме того, для данного языка существует множество подключаемых библиотек, предоставляющих готовые решения управления данными баз данных различных видов, включая PostgreSQL [17].

2.6.2 Описание операционной системы

Операционная система – совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг [18]. В качестве операционной системы используется Windows 10.

Компонентами пользовательского интерфейса Windows 10 являются окна с элементами управления, панель задач и контекстное меню, а основными устройствами ввода являются мышь и клавиатура.

2.6.3 Описание среды разработки

В качестве среды программирования используется Visual Studio 2019. Она включает в себя редактор исходного кода, поддерживающий технологию анализа кода IntelliSense и предоставляющий возможность простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня [19]. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения

2.6.4 Описание системы управления базами данных

Система управления базами данных – совокупность программных и языковых средств, обеспечивающих управление базами данных [14]. При написании плагина для хранения данных о коричневых знаках используется СУБД PostgreSQL.

PostgreSQL – это объектно-реляционная система управления базами данных. Данная СУБД является одной из наиболее популярных свободно распространяемых СУБД с открытым исходом кодом [20].

В PostgreSQL используется процедурный язык PL/pgSQL, этот язык позволяет сгруппировать блок вычислений и последовательность запросов внутри сервера базы данных, таким образом, мы получаем силу

процедурного языка и простоту использования SQL при значительной экономии накладных расходов на клиент-серверное взаимодействие. Благодаря этому, исключаются дополнительные обращения между клиентом и сервером, промежуточные ненужные результаты не передаются между сервером и клиентом, а также есть возможность избежать многочисленных разборов одного запроса

3 Реализация системы

3.1 Разработка и описание интерфейса пользователя

После авторизации пользователя в системе ему становится доступной карта местности, на которую он может добавлять знаки дорожного движения, в том числе со знаками типа «Коричневые знаки». Изображение знака создается в окне «Редактор знака». Для того, чтобы открыть это окно, необходимо перейти по ссылке «Создать» в окне создания знака. Последовательность переходов к форме «Редактор знака» отмечена красным цветом на рисунке 33.

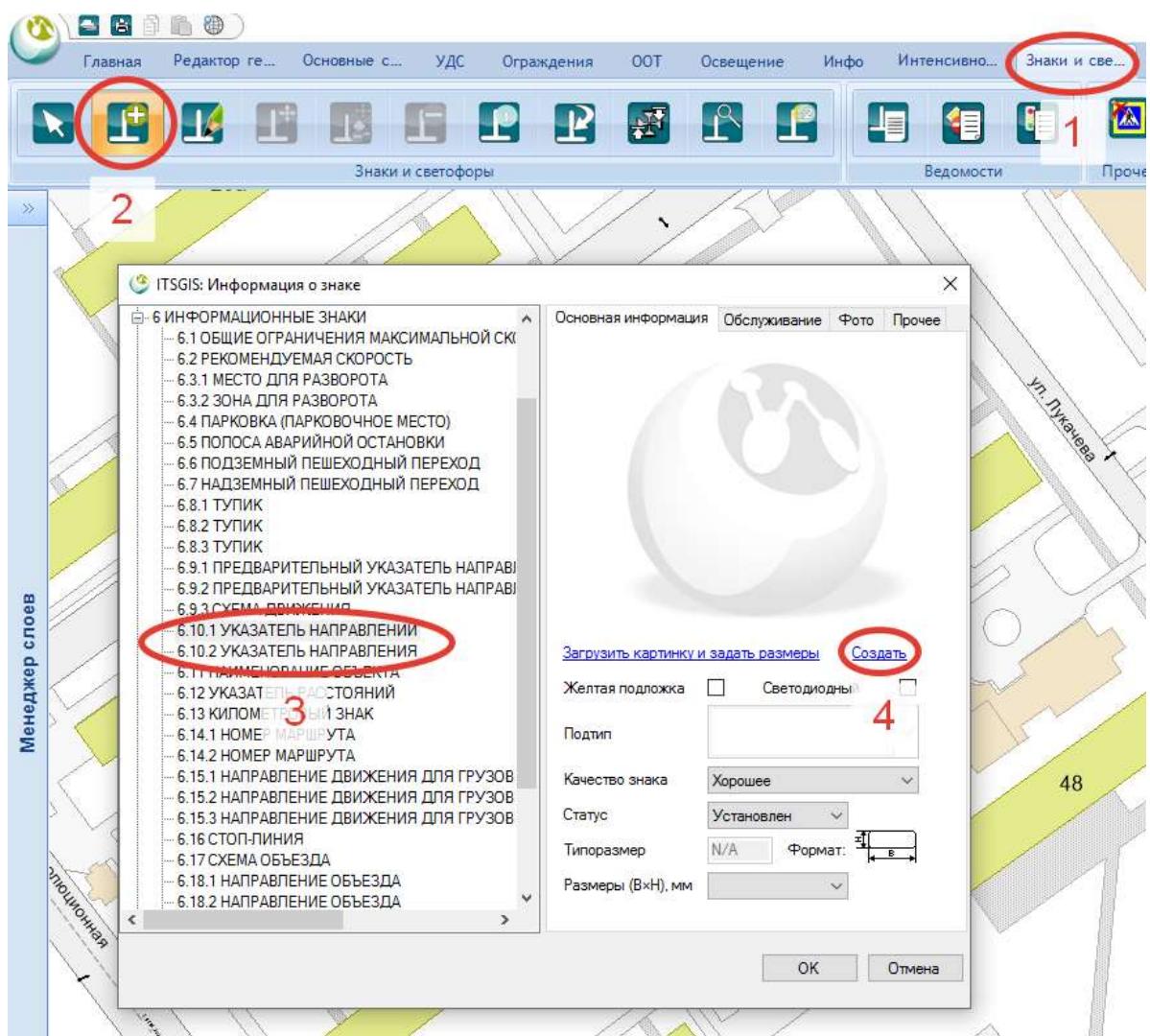


Рисунок 33 – Последовательность переходов к форме «Редактор знака»

На рисунке 34 представлен фрагмент формы редактора знака с пользовательским меню.

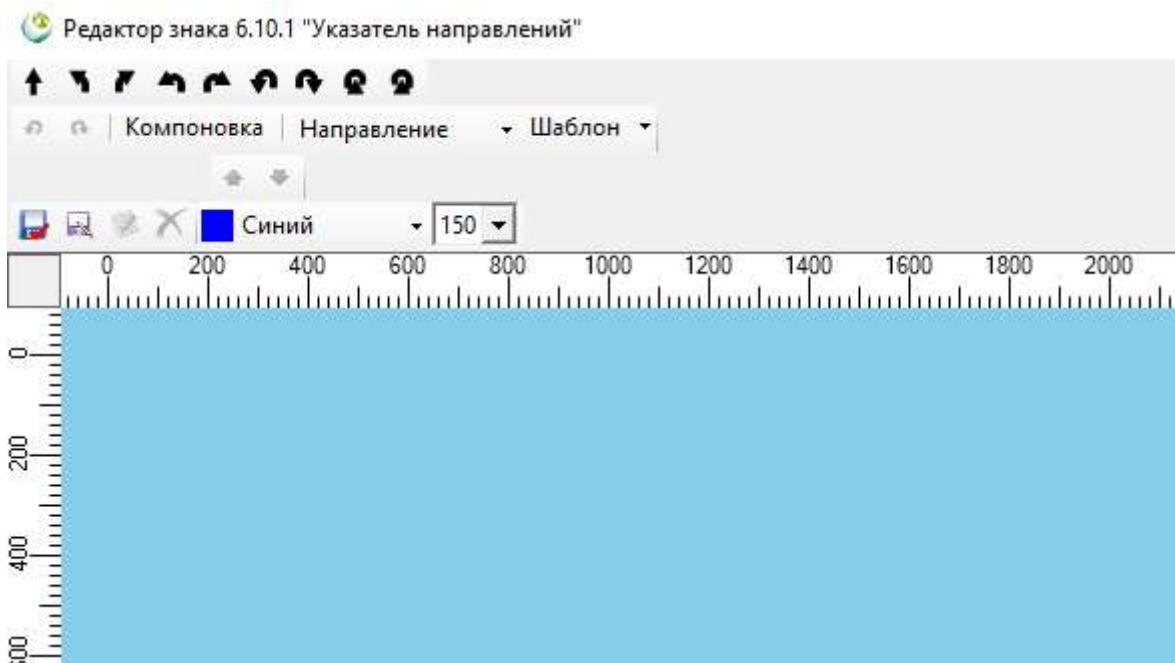


Рисунок 34 – Фрагмент экранной формы редактора знака (с меню)

В таблице 4 представлено описание элементов меню экранной формы редактора знака.

Таблица 4 – Описание элементов меню экранной форму редактора знака

Элемент	Описание
1	2
	Панель стрелок. Для добавления стрелки направления на знак, необходимо нажать на, любую из представленных, пиктограмму стрелки, и затем нажать на левый край знака – изображение стрелки добавится на знак или заменится на изображение выбранной стрелки (в случае, если на знаке уже есть стрелка)
	Стрелки перемещения элемента знака «Направление». По нажатию на соответствующую кнопку элемент перемещает на уровень выше или ниже

Продолжение таблицы 4

1	2
Компоновка	По нажатию этой кнопки открывается диалоговое окно «Компоновка знака», в котором задаются горизонтальные и вертикальные расстояние между элементами знака. Диалоговое окно представлено на рисунке 35
Направление ▾	Комбобокс выбора элемента знака, этот элемент возможно добавить на знак. Доступны три варианта выбора: «Направление» (добавление стрелки на знак), «Надпись» (добавление надписи и расстояния, например, «Самара 0,5»), «Вставка» (добавление знака на знак)
Шаблон ▾	Комбобокс, состоит из двух вариантов выбора – «Загрузить» и «Сохранить». При выборе «Загрузить» открывается диалоговое окно выбора файла шаблона знака, при выборе «Сохранить» –диалоговое окно сохранения файла шаблона знака
	Кнопки возврата и отмены последнего изменения знака. Например, по нажатию отменяется добавление стрелки на знак или возвращается
	Сохранение изображения знака в текущем файле в формате .dxf. DXF (англ. Drawing eXchange Format) – открытый формат файлов для обмена графической информацией между приложениями автоматизированного проектирования
Белый ▾	Комбобокс выбора цвета выделенного компонента знака. Возможен выбор следующих цветов: «Белый», «Зелёный», «Синий», «Коричневый». Цвета доступны только для элементов типа «Надпись» и «Вставка»

Продолжение таблицы 4

1	2
	По нажатию на кнопку удаляется выбранный элемент знака
	Кнопка открывает диалоговое окно сохранения изображения знака в новом файле формата .dxf
150	Комбобокс выбора высоты прописной буквы. Высота прописной буквы одинакова для всех надписей на знаке. Возможно следующие варианты выбора: 75, 100, 150, 200, 250, 300, 400 и 500

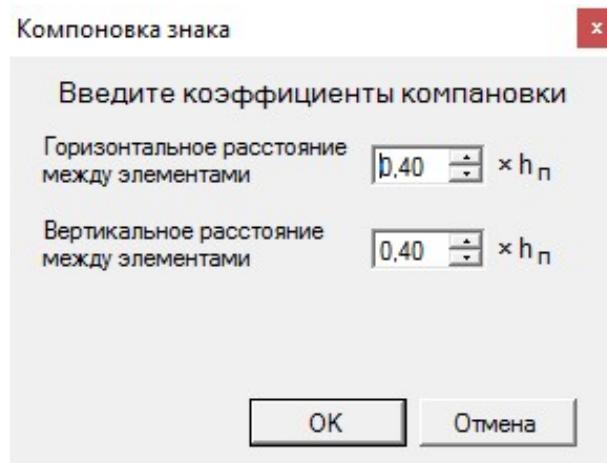


Рисунок 35 – Диалоговое окно «Компоновка знака»

3.1.1 Описание тестового примера

Создадим изображение знака. Для этого:

- 1) нажмём на **Направление** , затем выберем и кликнем по полю знака (на знак добавится изображение стрелки на знак);
- 2) нажмём на **Надпись** , кликнем правее изображения стрелки на знаке, затем введём надпись «ПОЛЯНКА» и зададим расстояние 33 км;
- 3) нажмём на **Вставка** , кликнем под надписью «ПОЛЯНКА 33», введём надпись вставки «музей ПЕТРОВА» и зададим расстояние 5 км (на знак добавится зелёная вставка с надписью «музей ПЕТРОВА 5»);
- 4) кликнем по зелёной вставке «музей ПЕТРОВА 5» (вставка

выделится оранжевым цветом), выберем  Коричневый (вставка станет коричневой).

Полученное изображение знака представлено на рисунке 36.

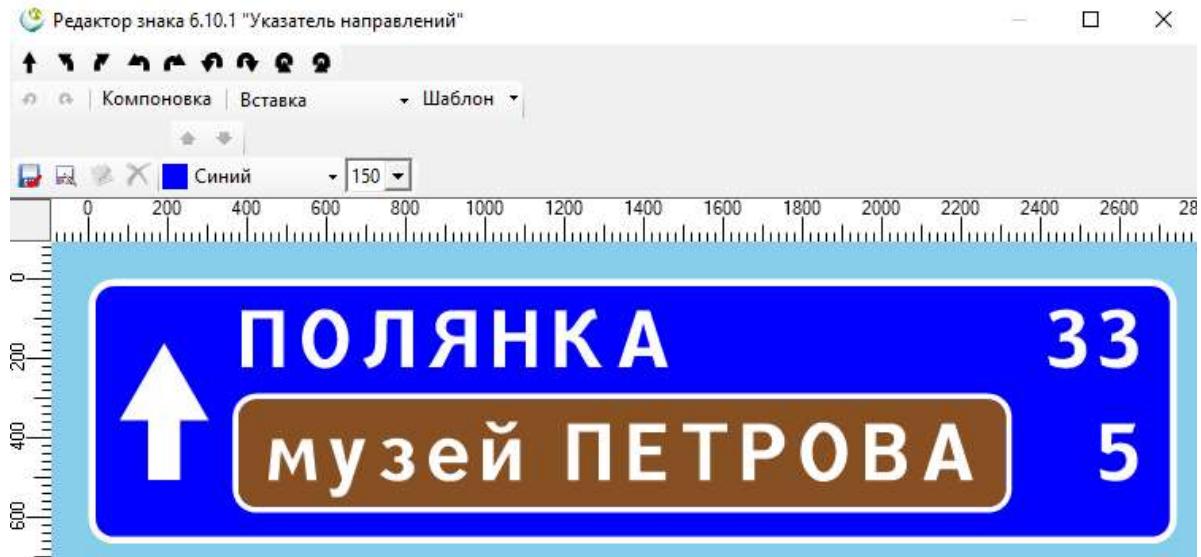


Рисунок 36 – Созданное изображение знака

3.2 Диаграммы реализации

Диаграммы реализации предназначены для отображения состава компилируемых и выполняемых модулей системы, а также связей между ними. Диаграммы реализации разделяются на два конкретных вида: диаграммы компонентов (component diagrams) и диаграммы развёртывания (deployment diagrams) [13].

3.2.1 Диаграмма компонентов

Диаграмма компонентов – диаграмма реализации, описывающая особенности физической реализации приложения, определяет архитектуру приложения и устанавливает зависимость между компонентами, в роли которых выступает исполняемый код. Диаграмма компонентов отображает общие зависимости между компонентами. Основными графическими элементами диаграммы являются компоненты, интерфейсы и зависимости между ними [13].

Диаграмма компонентов системы представлена на рисунке 37.

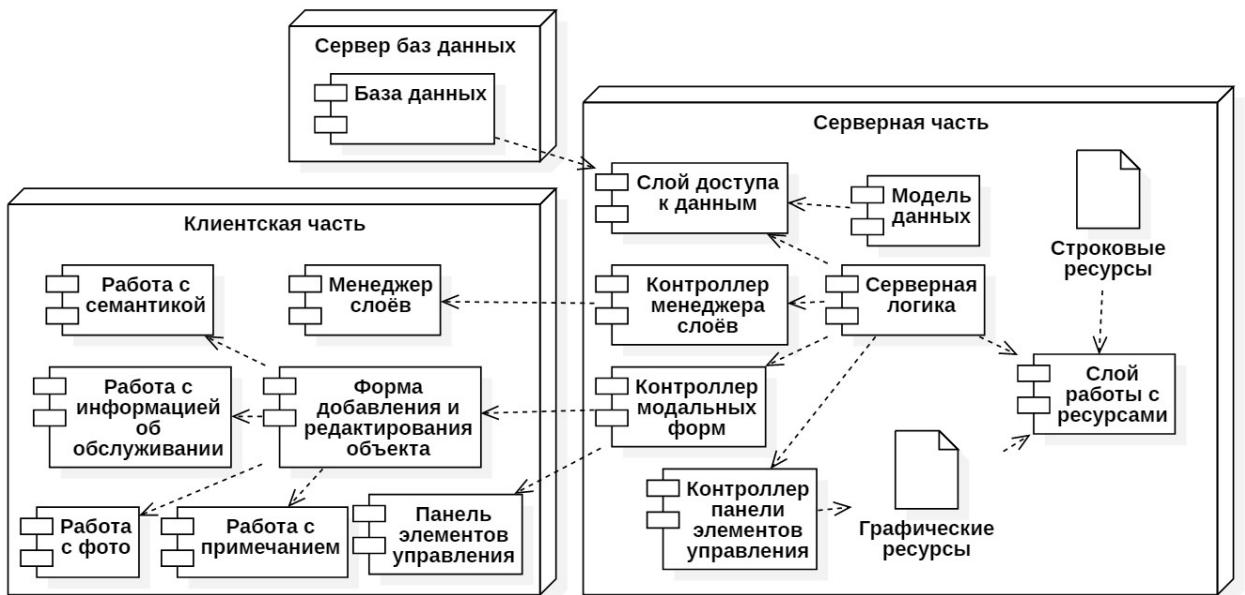


Рисунок 37 – Диаграмма компонентов системы с интегрированным плагином «Коричневые знаки»

3.2.2 Диаграмма развертывания

Диаграмма развертывания представляет физическое расположение системы, показывая, на каком физическом оборудовании запускается та или иная составляющая программного обеспечения [13].

Главными элементами диаграммы являются узлы, связанные информационными путями. Узел – это то, что может содержать программное обеспечение. Узлы бывают двух типов. Устройство – это физическое оборудование: компьютер или устройство, связанное с системой. Среда выполнения – это программное обеспечение, которое само может включать другое программное обеспечение, например, операционную систему.

Информационные пути между узлами представляют обмен информацией в системе. Эти пути могут сопровождаться информацией об используемых информационных протоколах [13].

Диаграмма развертывания представлена на рисунке 38.

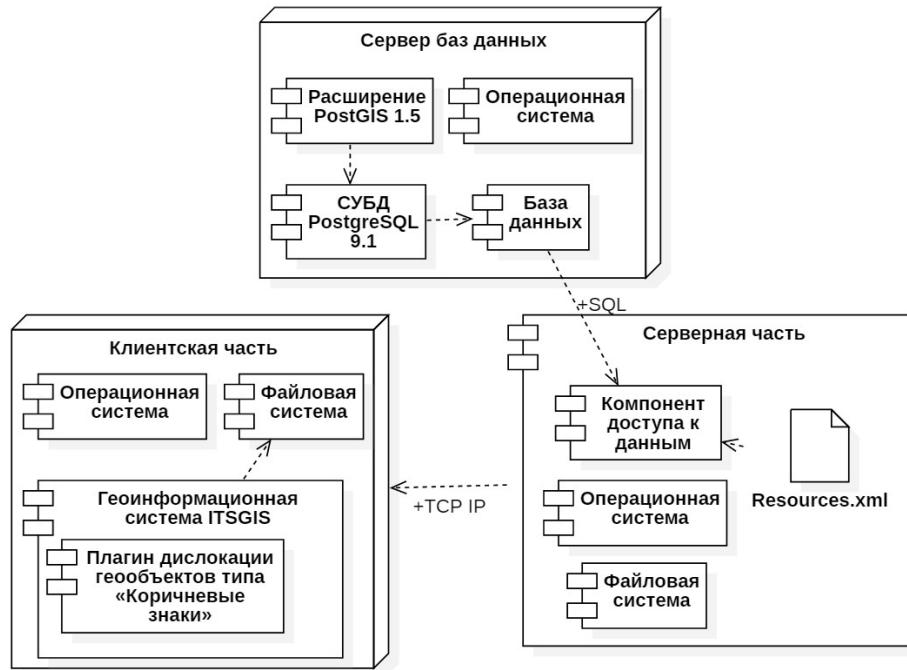


Рисунок 38 – Диаграмма развёртывания

3.2.3 Диаграмма классов

В соответствии со спецификациями, указанными в пункте 2.3, и выбором комплекса программных средств, указанным в пункте 2.6, были созданы классы с различными параметрами и методами.

На рисунке 39 приведена диаграмма классов подсистемы изображения расстояния между компонентами знака (этап реализации).

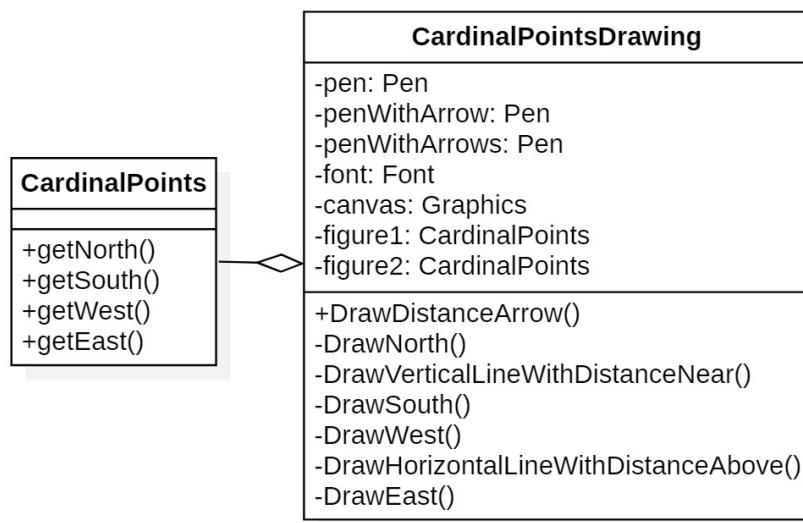


Рисунок 39 – Диаграмма классов подсистемы изображения расстояния между компонентами знака

3.3 Выбор и обоснование комплекса технических средств

3.3.1 Расчёт объёма занимаемой памяти клиентской части системы

Расчет объема внешней памяти

Для расчёта необходимого объёма свободной внешней памяти, необходимой для функционирования системы, воспользуемся следующей формулой:

$$V_{\text{ЖД}} = V_{\text{ОС}} + V_{\text{ITSGIS}} + V_{\Phi},$$

где $V_{\text{ОС}}$ – объём памяти, занимаемый операционной системой (операционная система Windows 10, $V_{\text{ОС}} = 32 \text{ Гб}$);

V_{ITSGIS} – объём памяти, занимаемый системой ITSGIS со встроенным в неё плагином «Коричневые знаки» ($V_{\text{ITSGIS}} = 264 \text{ Мб}$).

V_{Φ} – объём памяти, занимаемый 1000 файлов изображений знаков (объём одного файла не превышает 50 Кб).

Таким образом, суммарный объём внешней памяти составит:

$$V_{\text{ЖД}} = 32 \text{ Гб} + 264 \text{ Мб} + 50 \text{ Мб} \approx 32,4 \text{ Гб.}$$

Расчёт объема ОЗУ

Для расчёта необходимого объема ОЗУ воспользуемся следующей формулой:

$$V_{\text{ОЗУ}} = V_{\text{ОС}} + V_{\text{ITSGIS}} + V_{\Phi},$$

где $V_{\text{ОС}}$ – объём ОЗУ, занимаемое операционной системой (не превышает 1 Гб);

V_{ITSGIS} – объём ОЗУ, занимаемый системой ITSGIS со встроенным в неё плагином «Коричневые знаки» (не превышает 100 Мб);

V_{Φ} – объём ОЗУ, занимаемый файлом изображения коричневого знака (не превышает 50 Кб).

Суммарные объемы ОЗУ составят:

$$V_{OZY} = 1 \text{ Гб} + 100 \text{ Мб} + 50 \text{ Кб} \approx 1,2 \text{ Гб.}$$

Таким образом, 1,2 Гб оперативной памяти можно счесть минимально необходимым для функционирования системы.

3.3.2 Минимальные требования, предъявляемые к системе изображения расстояния между компонентами знака

Для корректного функционирования системы необходимо:

- 1) тип ЭВМ: x86-64 совместимый;
- 2) объем ОЗУ – не менее 1,2 Гб;
- 3) объем свободного дискового пространства – не менее 32,4 Гб;
- 4) клавиатура или иное устройство ввода;
- 5) мышь или иное манипулирующее устройство;
- 6) геоинформационная система ITSGIS версии 2.0.1.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта была разработана автоматизированная система изображения расстояния между компонентами знака, позволяющая отмечать расстояния между компонентами знака на изображении знака.

В первом разделе приведены основные понятия предметной области, характеристики систем-аналогов и результаты их сравнительного анализа, на основании этого выполнена объектная декомпозиция, отраженная в диаграмме объектов. Сформулирована постановка задачи.

Во втором разделе описана общая структура ГИС ITSGIS, структурная схема системы коричневых знаков, в рамках которой была разработана подсистема изображения расстояния между компонентами знака, описаны прототипы экранных форм. Также был разработан логический проект, в который вошли все канонически диаграммы. Описан комплекс программных средств.

В третьем разделе приведено описание пользовательского меню, текстового примера и приведены диаграммы реализации. Также были проведены ресурсные расчёты и определены технические требования для функционирования клиентской части системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 О правилах дорожного движения (с изменениями на 31 декабря 2020 года). Собрание актов Президента и Правительства Российской Федерации, 1993, N 47, ст.4531
- 2 Кузнецова Анна Петровна История дорожных знаков // ООО «ИндорСофт». 2014. №1 (2). URL: <https://cyberleninka.ru/article/n/istoriya-dorozhnyh-znakov> (дата обращения: 16.11.2020).
- 3 ГОСТ Р 52290-2004 Технические средства организации дорожного движения. Знаки дорожные. Общие технические требования (с Поправками, с Изменениями N 1, 2, 3). М.: Стандартинформ, 2006. 229 с.
- 4 Проекты Яндекса в России – Яндекс.Радар [Электронный ресурс]. URL: <https://radar.yandex.ru/yandex?month=2020-10> (дата обращения: 16.11.2020).
- 5 Картографический сервис 2ГИС становится частью экосистемы Сбербанка [Электронный ресурс]. URL: <https://info.2gis.ru/moscow/company/news/kartograficheskiy-servis-2gis-stanovitsya-chastyu-ekosistemy-sberbanka> (дата обращения: 16.11.2020).
- 6 Alphabet Investor Relations [Электронный ресурс]. URL: <https://abc.xyz/investor/> (дата обращения: 16.11.2020).
- 7 Яндекс.Карты [Электронный ресурс]. URL: <https://yandex.ru/maps/> (дата обращения: 16.11.2020).
- 8 Карта городов России [Электронный ресурс]. URL: <https://2gis.ru/> (дата обращения: 16.11.2020).
- 9 Google Карты [Электронный ресурс]. URL: <https://www.google.com/maps> (дата обращения: 16.11.2020).
- 10 Программная инженерия. Учебник для вузов. 5-е издания обновлённое и дополненное. Стандарт третьего поколения. – СПб.: Питер, 2016. – 640 с.: ил. – (Серия «Учебник для вузов»).
- 11 Геоинформационная система «ITSGIS» [Электронный ресурс]. URL: <http://itsgis.ru/about> (дата обращения: 19.11.2020).

12 УМЛ. Проектирование систем реального времени, параллельных и распределённых приложений: Пер. с англ. М.: ДМК Пресс, 2016. 700 с.: ил. (Серия «Объектно-ориентированные технологии в программировании»).

13 Буч, Г. Язык UML. Руководство пользователя [Текст] /Г. Буч, Д. Рамбо, А. Якобсон. 2-е изд.: Пер. с англ. Мухина Н. М.: ДМК Пресс, 2006. 496 с.: ил.

14 Введение в реляционные базы данных / В. В. Кириллов, Г. Ю. Громов. СПб.: БХВ-Петербург, 2009. 464 с.: ил. + CD-ROM (Учебная литература для вузов).

15 ГОСТ 28806-90. Качество программных средств. Термины и определения. М., 1990. 12 с. (Издательство стандартов).

16 ГОСТ 28397-89. Языки программирования. Термины и определения М., 1989. 8 с. (Издательство стандартов).

17 Npgsql Documentation [Электронный ресурс]. URL: <https://www.npgsql.org/doc/index.html> (дата обращения: 25.11.2020).

18 ГОСТ 15971-90. Системы обработки информации. Термины и определения. М.: Издательство стандартов. 1991. 14 с.

19 IDE Visual Studio 2019 [Электронный ресурс]. URL: <https://visualstudio.microsoft.com/ru/vs/> (дата обращения: 25.11.2020).

20 DB-Engines Ranking [Электронный ресурс]. URL: <https://db-engines.com/en/ranking> (дата обращения: 01.12.2020).

Приложение А

Руководство пользователя

A.1 Назначение системы

Программа позволяет создать тематический слой геообъектов «Коричневые знаки» на электронной карте города, а также предоставляет возможность добавления, выделения, редактирования и удаления геообъектов, группировку геообъектов по тематическим слоям. В системе имеется возможность сохранения данных о коричневых знаках и их характеристиках в базу данных. Данные о лестницах (тип, размер, место расположения и т.д.) хранятся в файлах типа *.sql.

A.2 Условия работы системы

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств:

1 Требования к техническому обеспечению:

- объём ОЗУ – не менее 1,2 Гб;
- объём свободного дискового пространства – не менее 32,4 Гб;
- клавиатура или иное устройство ввода;
- мышь или иное манипулирующее устройство;
- процессор Intel или AMD с частотой не менее 1,8 ГГц;
- монитор с разрешающей способностью не ниже 1280 x 1024.

2 Требования к программному обеспечению:

- операционная система Windows 10;
- установленная СУБД PostgreSQL 9.1.

A.3 Установка системы

Система поставляется в виде zip-архива. Данный файл необходимо распаковать в любую директорию на жестком диске. Запускаемым файлом системы является файл ITSGIS.exe

A.4 Работа с системой

После авторизации пользователя в системе ему становится доступной карта местности, на которую он может добавлять знаки дорожного движения, в том числе со знаками типа «Коричневые знаки».

Изображение знака создаётся в окне «Редактор знака». Для того, чтобы открыть это окно, необходимо перейти по ссылке «Создать» в окне создания знака. Последовательность переходов к форме «Редактор знака» отмечена красным цветом на рисунке 1.

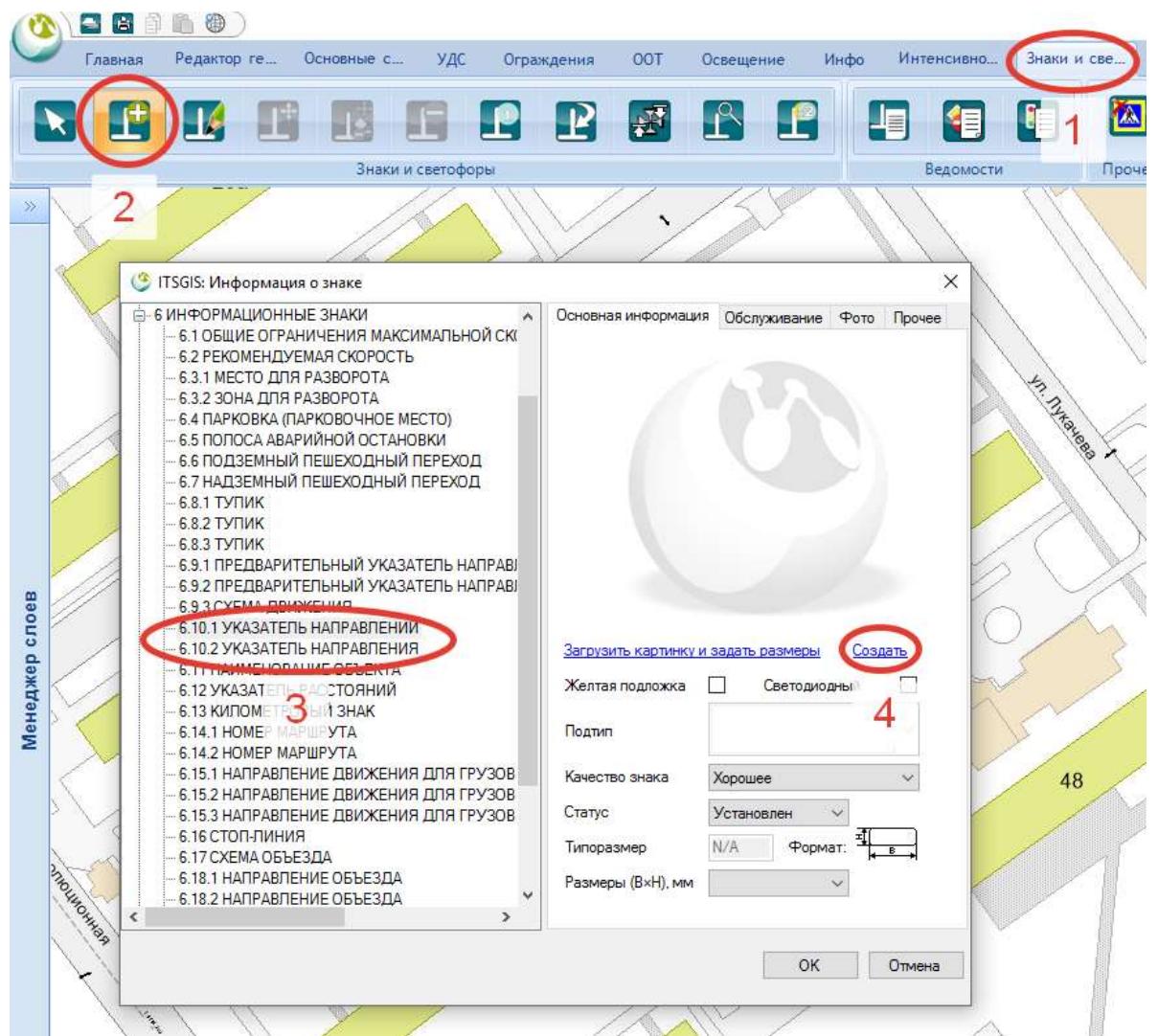


Рисунок 1 – Последовательность переходов к форме «Редактор знака»

На рисунке 2 представлен фрагмент формы редактора знака с пользовательским меню.

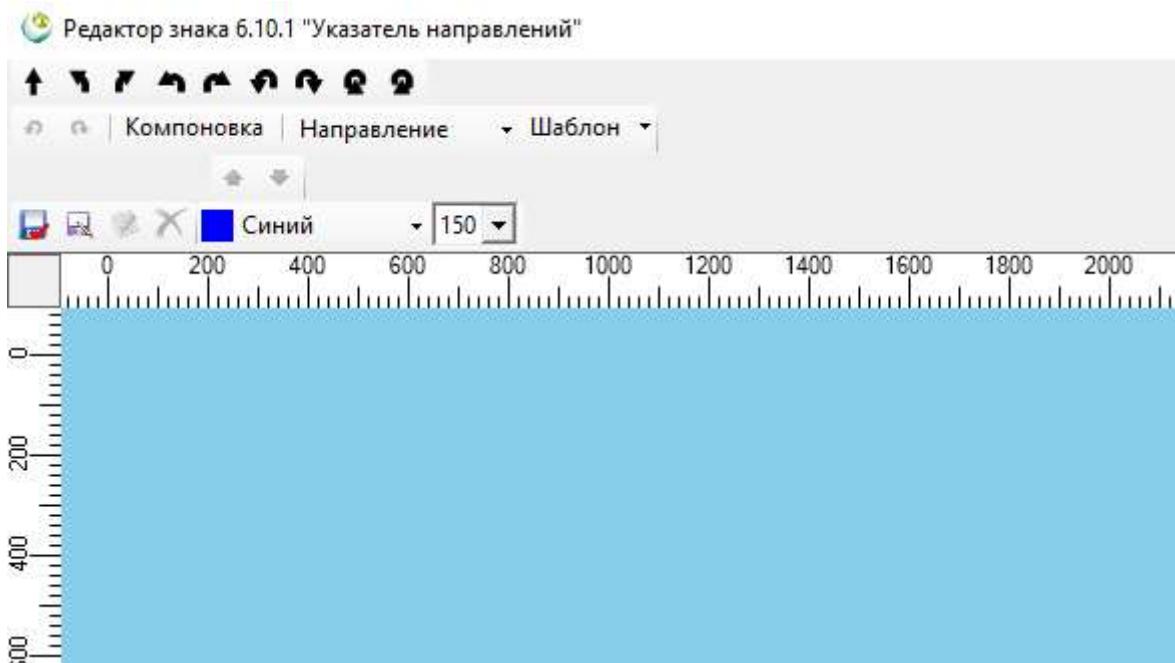


Рисунок 2 – Фрагмент экранной формы редактора знака (с меню)

В таблице 1 представлено описание элементов меню экранной формы редактора знака.

Таблица 1 – Описание элементов меню экранной формы редактора знака

Элемент	Описание
1	2
	Панель стрелок. Для добавления стрелки направления на знак, необходимо нажать на, любую из представленных, пиктограмму стрелки, и затем нажать на левый край знака – изображение стрелки добавится на знак или заменится на изображение выбранной стрелки (в случае, если на знаке уже есть стрелка)
	Стрелки перемещения элемента знака «Направление». По нажатию на соответствующую кнопку элемент перемещает на уровень выше или ниже

Продолжение таблицы 1

1	2
Компоновка	По нажатию этой кнопки открывается диалоговое окно «Компоновка знака», в котором задаются горизонтальные и вертикальные расстояние между элементами знака. Диалоговое окно представлено на рисунке 3
Направление ▾	Комбобокс выбора элемента знака, этот элемент возможно добавить на знак. Доступны три варианта выбора: «Направление» (добавление стрелки на знак), «Надпись» (добавление надписи и расстояния, например, «Самара 0,5»), «Вставка» (добавление знака на знак)
Шаблон ▾	Комбобокс, состоит из двух вариантов выбора – «Загрузить» и «Сохранить». При выборе «Загрузить» открывается диалоговое окно выбора файла шаблона знака, при выборе «Сохранить» –диалоговое окно сохранения файла шаблона знака
↶ ↷	Кнопки возврата и отмены последнего изменения знака. Например, по нажатию отменяется добавление стрелки на знак или возвращается
	Сохранение изображения знака в текущем файле в формате .dxf. DXF (англ. Drawing eXchange Format) – открытый формат файлов для обмена графической информацией между приложениями автоматизированного проектирования
Белый ▾	Комбобокс выбора цвета выделенного компонента знака. Возможен выбор следующих цветов: «Белый», «Зелёный», «Синий», «Коричневый». Цвета доступны только для элементов типа «Надпись» и «Вставка»

Продолжение таблицы 1

1	2
	По нажатию на кнопку удаляется выбранный элемент знака
	Кнопка открывает диалоговое окно сохранения изображения знака в новом файле формата .dxf
150	Комбобокс выбора высоты прописной буквы. Высота прописной буквы одинакова для всех надписей на знаке. Возможно следующие варианты выбора: 75, 100, 150, 200, 250, 300, 400 и 500

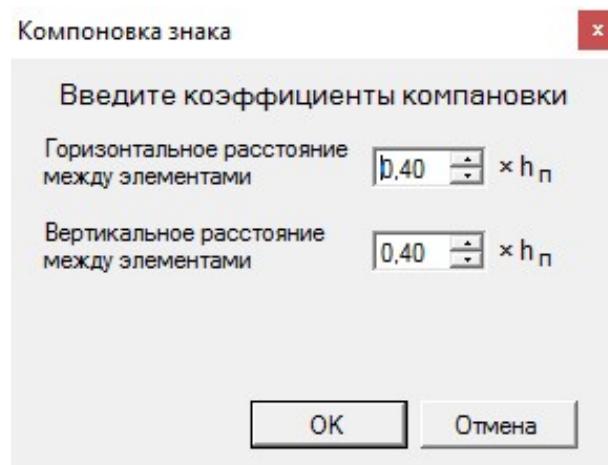


Рисунок 3 – Диалоговое окно «Компоновка знака»

Приложение Б

Листинг модулей программы

Класс CardinalPoints:

```
using System;
using System.Drawing;

namespace ITS.Common.SpecialSignEditor.Drawing
{
    // точки, соответствующие сторонам света:
    public class CardinalPoints
    {
        private readonly PointF northPoint;
        private readonly PointF southPoint;
        private readonly PointF westPoint;
        private readonly PointF eastPoint;

        public CardinalPoints(PointF[] points)
        {
            if (points == null)
                throw new NullReferenceException();

            northPoint = points[0];
            southPoint = points[0];
            westPoint = points[0];
            eastPoint = points[0];

            foreach (var point in points)
            {
                if (point.Y < northPoint.Y)
                {
                    northPoint = point;
                }

                if (point.Y > southPoint.Y)
                {
                    southPoint = point;
                }

                if (point.X < westPoint.X)
                {
                    westPoint = point;
                }

                if (point.X > eastPoint.X)
                {
                    eastPoint = point;
                }
            }
        }

        public PointF North => northPoint;
        public PointF South => southPoint;
        public PointF West => westPoint;
        public PointF East => eastPoint;
    }
}
```

Класс CardinalPointsDrawing:

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;

namespace ITS.Common.SpecialSignEditor.Drawing
{
    public static class CardinalPointsDrawing
    {
        public static int TextHeight = 0;
        private static int DistanceCoeff = TextHeight;

        private static readonly Pen Pen = new Pen(Color.Black, 11);
        private static readonly Pen PenWithArrow = new Pen(Pen.Color, Pen.Width)
        {
            StartCap = LineCap.ArrowAnchor,
        };
        private static readonly Pen PenWithArrows = new Pen(Pen.Color, Pen.Width)
        {
            StartCap = LineCap.ArrowAnchor,
            EndCap = LineCap.ArrowAnchor
        };

        //new Font(SpecialSign.FontFamily, text.TextHeight, FontStyle.Bold)
        private static readonly Font Font = new Font("Arial", 100);
        private static readonly SolidBrush Brush = new SolidBrush(Pen.Color);

        // выделяю память один раз, чтобы было меньше работы для сборщика мусора:
        private static PointF start;
        private static PointF end;
        private static PointF left;
        private static PointF right;
        private static PointF up;
        private static PointF low;

        public static void DrawDistanceArrow(Graphics gr, PointF[] fig1Points, PointF[]
fig2Points)
        {
            // поиск точек, соответствующих сторонам света:
            CardinalPoints fig1Cps = new CardinalPoints(fig1Points);
            CardinalPoints fig2Cps = new CardinalPoints(fig2Points);

            bool isNorthFig2 = fig1Cps.North.Y > fig2Cps.South.Y;
            bool isSouthFig2 = fig1Cps.South.Y < fig2Cps.North.Y;
            bool isWestFig2 = fig1Cps.West.X > fig2Cps.East.X;
            bool isEastFig2 = fig1Cps.East.X < fig2Cps.West.X;

            if (isNorthFig2)
                DrawNorth(gr, fig1Cps, fig2Cps);
            else if (isSouthFig2)
                DrawSouth(gr, fig1Cps, fig2Cps);
            else if (isWestFig2)
                DrawWest(gr, fig1Cps, fig2Cps);
            else if (isEastFig2)
                DrawEast(gr, fig1Cps, fig2Cps);
        }

        private static void DrawNorth(Graphics gr, CardinalPoints fig1Cps, CardinalPoints
fig2Cps)
        {
            start.X = fig1Cps.North.X;
            start.Y = fig1Cps.North.Y;

            end.X = start.X;
```

```

end.Y = fig2Cps.South.Y;

DrawVerticalLineWithDistanceNear(gr, start, end);

gr.DrawLine(Pen,
    fig1Cps.West.X, fig1Cps.North.Y,
    fig1Cps.East.X, fig1Cps.North.Y);

// ДОВОДИТЬ ЛИНИЮ, если не хватает её длины:
left.X = fig2Cps.West.X > end.X ? end.X : fig2Cps.West.X;
left.Y = fig2Cps.South.Y;
// ДОВОДИТЬ ЛИНИЮ, если не хватает её длины:
right.X = fig2Cps.East.X < end.X ? end.X : fig2Cps.East.X;
right.Y = left.Y;

gr.DrawLine(Pen, left, right);
}

private static void DrawVerticalLineWithDistanceNear(Graphics gr, PointF
lineStart, PointF lineEnd)
{
    float lineLen = Math.Abs(lineEnd.Y - lineStart.Y);

    if (lineLen > Font.GetHeight())
    {
        gr.DrawLine(PenWithArrows, lineStart, lineEnd);
    }
    else
    {
        gr.DrawLine(Pen, lineStart, lineEnd);
        gr.DrawLine(PenWithArrow,
            lineStart.X, lineStart.Y,
            lineStart.X, lineStart.Y - Font.GetHeight());
        gr.DrawLine(PenWithArrow,
            lineEnd.X, lineEnd.Y,
            lineEnd.X, lineEnd.Y + Font.GetHeight());
    }

    int distance = (int)Math.Abs(lineEnd.Y - lineStart.Y);
    if (lineLen < Font.GetHeight() / 2f)
    {
        gr.DrawString(distance + " px", Font, Brush,
            lineStart.X + Font.Size / 4f,
            lineEnd.Y + Font.GetHeight() / 4f);
    }
    else
    {
        gr.DrawString(distance + " px", Font, Brush,
            lineStart.X + Font.Size / 4f,
            (lineStart.Y + lineEnd.Y) / 2f - Font.GetHeight() / 2f);
    }
}

private static void DrawSouth(Graphics gr, CardinalPoints fig1Cps, CardinalPoints
fig2Cps)
{
    start.X = fig1Cps.South.X;
    start.Y = fig1Cps.South.Y;

    end.X = start.X;
    end.Y = fig2Cps.North.Y;

    DrawVerticalLineWithDistanceNear(gr, start, end);

    gr.DrawLine(Pen,

```

```

        fig1Cps.West.X, fig1Cps.South.Y,
        fig1Cps.East.X, fig1Cps.South.Y);

    // доводить линию, если не хватает её длины:
    left.X = fig2Cps.West.X > end.X ? end.X : fig2Cps.West.X;
    left.Y = fig2Cps.North.Y;
    // доводить линию, если не хватает её длины:
    right.X = fig2Cps.East.X < end.X ? end.X : fig2Cps.East.X;
    right.Y = left.Y;

    gr.DrawLine(Pen, left, right);
}

private static void DrawWest(Graphics gr, CardinalPoints fig1Cps, CardinalPoints
fig2Cps)
{
    start.X = fig1Cps.West.X;
    start.Y = fig1Cps.West.Y;

    end.X = fig2Cps.East.X;
    end.Y = start.Y;

    DrawHorizontalLineWithDistanceAbove(gr, start, end);

    gr.DrawLine(Pen,
        fig1Cps.West.X, fig1Cps.North.Y,
        fig1Cps.West.X, fig1Cps.South.Y);

    // доводить линию, если не хватает её длины:
    up.X = fig2Cps.East.X;
    up.Y = fig2Cps.North.Y > end.Y ? end.Y : fig2Cps.North.Y;
    // доводить линию, если не хватает её длины:
    low.X = up.X;
    low.Y = fig2Cps.South.Y < end.Y ? end.Y : fig2Cps.South.Y;

    gr.DrawLine(Pen, up, low);
}

private static void DrawHorizontalLineWithDistanceAbove(Graphics gr, PointF
lineStart, PointF lineEnd)
{
    float lineLen = Math.Abs(lineEnd.X - lineStart.X);

    if (lineLen > Font.Size)
    {
        gr.DrawLine(PenWithArrows, lineStart, lineEnd);
    }
    else
    {
        gr.DrawLine(Pen, lineStart, lineEnd);
        gr.DrawLine(PenWithArrow,
            lineStart.X, lineStart.Y,
            lineStart.X - Font.Size * 2f, lineStart.Y);
        gr.DrawLine(PenWithArrow,
            lineEnd.X, lineEnd.Y,
            lineEnd.X + Font.Size * 2f, lineEnd.Y);
    }

    string distanceStr = (int)Math.Abs(lineEnd.X - lineStart.X) + " px";
    if (distanceStr.Length * Font.Size / 3f > lineLen)
    {
        gr.DrawString(distanceStr, Font, Brush,
            lineEnd.X + Font.Size / 3f,
            lineStart.Y - Font.GetHeight());
    }
}

```

```

    else
    {
        gr.DrawString(distanceStr, Font, Brush,
                     (lineStart.X + lineEnd.X) / 2f - distanceStr.Length / 3f * Font.Size,
                     lineStart.Y - Font.GetHeight());
    }
}

private static void DrawEast(Graphics gr, CardinalPoints fig1Cps, CardinalPoints
fig2Cps)
{
    start.X = fig1Cps.East.X;
    start.Y = fig1Cps.East.Y;

    end.X = fig2Cps.West.X;
    end.Y = start.Y;

    DrawHorizontalLineWithDistanceAbove(gr, start, end);

    gr.DrawLine(Pen,
                fig1Cps.East.X, fig1Cps.North.Y,
                fig1Cps.East.X, fig1Cps.South.Y);

    // ДОВОДИТЬ ЛИНИЮ, если не хватает её длины:
    up.X = fig2Cps.West.X;
    up.Y = fig2Cps.North.Y > end.Y ? end.Y : fig2Cps.North.Y;
    // ДОВОДИТЬ ЛИНИЮ, если не хватает её длины:
    low.X = up.X;
    low.Y = fig2Cps.South.Y < end.Y ? end.Y : fig2Cps.South.Y;

    gr.DrawLine(Pen, up, low);
}
}

```