Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Базовая кафедра геоинформационных систем

УТВЕРЖДАЮ
Заведующий кафедрой
_____ В.И. Харук
« _____» _____2018 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Методическое обеспечение разработки мобильных и веб-приложений картографической тематики

09.04.01 Информатика и вычислительная техника
09.04.01.07 Дистанционное зондирование и ГИС-технологии в мониторинге природных и антропогенных экосистем

Научный руководитель _____ _____ А.А. Гостева

Выпускник _____ Н.И. Рукосуева

Рецензент _____ _____ А.В. Токарев

Нормоконтролер _____ Е.В. Федотова

Красноярск 2018

СОДЕРЖАНИЕ

Введение 4

1 Общие рекомендации 6

1.1 Список определений и сокращений, необходимых для разработки веб- и мобильных ГИС 6

1.2 Классификация веб-сервисов. ГИС-сервисы 10

1.3 Примеры применения ГИС-технологий в веб-проектах 11

1.4 Платформы для веб- и мобильных ГИС 16

1.5 Минимальные требования к персональному компьютеру для разработки ГИС-приложений 17

2 Способы представления данных в ГИС 19

2.1 Правила выбора проекции 20

2.2 Форматы хранения пространственных данных 22

2.3 Стандарты ГИС-сервисов Open Geospatial Consortium 23

3 ГИС-серверы: обзор и тестирование производительности 25

3.1 Обзор ГИС-серверов 25

3.2 Сравнительный анализ ГИС-серверов 31

4 ГИС-сервисы: обзор и сравнение функционала 34

4.1 Обзор ГИС-сервисов 34

4.2 Сравнительный анализ ГИС-сервисов 37

5 Базы данных для хранения пространственных данных 41

5.1 Обзор систем управления базами данных 41

5.2 Тестирование СУБД 43

5.3 Результаты тестирования СУБД 44

6 Разработка мобильной ГИС 46
6.1 Исходный пакет инструментов для разработки мобильных ГИС 46
6.2 Инструменты пространственного анализа в мобильных ГИС 50
6.3 Реализация мобильной ГИС 52
7 Разработка веб-ГИС 58
7.1 Исходный пакет инструментов для разработки веб-ГИС 58
7.2 Языковые средства программирования для разработки веб-ГИС 59
7.3 Особенности UI/UX-проектирования и конструирования клиентской части веб-ГИС 60
7.4 Инструменты пространственного анализа в веб-ГИС 62
7.5 Реализация веб-ГИС 63
Заключение 72
Список использованных источников 73
Приложения А – Д 78-102

ВВЕДЕНИЕ

В последнее время все большую популярность приобретает использование сети Интернет для получения какой-либо услуги, выгоды, знаний и т.д. Удобство множества услуг, доступных дистанционно, трудно переоценить – это не только экономия времени, но и возможность более подробно изучить предлагаемый ассортимент товаров, услуг, сервисов. Также немаловажным является то, что воспользоваться всеми сайтами и веб-приложениями, что размещены в сети Интернет, можно воспользоваться в любое время суток.

Таким образом, для привлечения большей аудитории, компании или частному лицу необходимо иметь собственный сайт или веб-приложение. В основном информация предоставляется в совокупности текста, таблиц, списков, изображений на веб-страницах. Иногда, в зависимости от вида информации, которую необходимо предоставить посетителю сайта, используются также и различные виды отображения пространственных данных. Для этого используется специальная область геоинформационных технологий – веб-картография.

Веб-картография является очень молодым направлением геоинформационных технологий. Первая электронная карта была представлена М.Я. Крааком в 2001 году, однако область веб-картографии начала популяризироваться примерно с 2003 года одновременно с появлением проекта NASA World Wind. Еще большую популярность в бизнесе и повседневной жизни данное направление начало приобретать в начале 2010 года с увеличением скорости Интернет-соединения на персональных компьютерах, а с развитием мобильных технологий и вовсе стало использоваться практически повсюду – от указания на карте местонахождения компании и сайтов с картами городов до мощных сервисов, позволяющих работать с пространственными данными прямо в браузере или мобильном приложении, причем набор инструментов таких сервисов не уступает аналогичным настольным приложениям.

В сети Интернет имеется много информации о веб-картографии, но она не является исчерпывающей, зачастую разрознена и требует большого количества времени на поиск и структурирование, что создает видимость высокого порога вхождения в данное направление. Необходимо собрать воедино основные аспекты создания мобильных и веб-приложений с использованием пространственных данных, которые позволят определиться с основными критериями и способами построения таких приложений. Это позволит уменьшить порог вхождения, способствуя еще большей популяризации веб-картографии в бизнесе и повседневной жизни.

Цель данной работы – провести исследование предметной области веб-картографии, выявить основные определения и термины. Определить характеристики инструментов, предлагаемых для создания мобильных и веб-приложений картографической тематики, сравнить их между собой. Составить перечень методических указаний по проектированию мобильных и веб-приложений, сконструировать универсальные мобильное и веб-приложения.

Задачи работы:

— обозначить общие определения и термины предметной области веб-картографии, выявить отрасли, в которых может использоваться отображение пространственных данных;

— определить технологии, необходимые для построения картографических мобильных и веб-приложений;

— провести исследование и сопоставить результаты тестирования технологий для построения картографических мобильных и веб-приложений;

— спроектировать и запрограммировать примеры мобильных и веб-приложений;

— приложить рекомендации для разработки картографических мобильных и веб-приложений для каждой платформы.

1 Общие рекомендации

На протяжении всей работы будут использоваться определения, на которые необходимо опираться при создании картографического веб или мобильного приложения. Список определений указан в разделе 1.1. Затем необходимо определить цель, для которой оно создается, область применения, в зависимости от этих параметров будут выбраны такие параметры, как картографическая проекция, платформа, на которой приложение будет запускаться и т.д.

1.1 Список определений и сокращений, необходимых для разработки веб- и мобильных ГИС

Для выполнения требований технических регламентов Евразийского экономического союза и оценки соответствия требованиям технических регламентов Союза на добровольной основе применяются межгосударственные и международные стандарты. Для определения однозначности терминов, использующихся в данной работе, их определения взяты из Межгосударственного стандарта (ГОСТ).

Автоматизированная система по ГОСТ 34.003-90 «Автоматизированные системы. Термины и определения» – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (по ГОСТ 20886-85).

Веб-интерфейс по ГОСТ Р 52872-2012 – интерфейс пользователя для работы с каким-либо элементом сервера в режиме онлайн.

Веб-приложение по ГОСТ Р ИСО 9241-151-2014 – приложение, предоставляющее функциональные возможности пользователю через браузер или другой тип агента пользователя, использующего веб-форматы и протоколы. Примечание – веб-приложения в контексте настоящего стандарта включают веб-сайты, которые только поставляют информационное наполнение, сочетают доставку контента с характерными для приложения функциональными возможностями, или предоставляют только определенные прикладные функциональные возможности, такие как конкретный веб-сервис.

Веб-сайт – совокупность веб-страниц, представляющих собой узел сети Интернет и объединенных по смыслу, навигационно и физически находящихся на одном веб-сервере (по ГОСТ Р 52872-2012).

Геоинформационная система; ГИС по ГОСТ 52438-2005 – информационная система, оперирующая пространственными данными.

Информационная система по ГОСТ 52438-2005 – система, предназначенная для хранения, обработки, поиска, распространения, передачи и представления информации.

Картографическая семиотика по ГОСТ 21667-76 – раздел картографии, разрабатывающий общую теорию систем картографических условных знаков и использования способов картографического изображения.

Картографическая топонимика по ГОСТ 21667-76 – раздел картографии, изучающий способы и правила написания географических названий на карте.

Программное обеспечение геоинформационной системы по ГОСТ 52438-2005 – совокупность программ, в которых реализованы функциональные возможности геоинформационных систем и сопровождающей программной документации.

Пространственные данные по ГОСТ Р 52155-2003 – цифровые данные о пространственных объектах, включающие сведения об их местоположении, форме и свойствах, представленные в координатно-временной системе.

Сервер по ГОСТ Р 52872-2012 – сетевой узел, содержащий данные и предоставляющий услуги другим запрашивающим станциям (например, другим компьютерам в сети).

Так как область Интернет-технологий относительно недавно начала широко развиваться, некоторые определения имеют не совсем полные формулировки по ГОСТ. Существует множество других источников (информационные, обучающие веб-сайты), содержащих трактовки таких определений. Некоторые формулировки определений целесообразно совмещать друг с другом, дополняя либо избавляя их от лишних уточнений. Таким образом, для следующих определений были подобраны и составлены наиболее полные и исчерпывающие понятия.

Бенчмарк – компьютерная программа, содержащая в себе набор функций, необходимых для сравнения производительности компьютеров и программных компонентов.

Бенчмаркинг – эталонное тестирование на основе бенчмарка.

Веб-ГИС – геоинформационная система, ориентированная на ее использование в компьютерных сетях типа Интернет или Интранет, обычно реализованная на ГИС-серверах.

Веб-картография – область компьютерных технологий, одно из направлений геоинформационных систем, связанное с доставкой пространственных данных конечному пользователю посредством веб-страниц [1].

Веб-портал – веб-сайт, представляющий собой совокупность нескольких сайтов либо страниц одного сайта, объединенных с целью предоставления доступа к интерактивным сервисам в рамках этой группы.

Веб-сервер – компьютер, предоставляющий компьютерное и программное обеспечение, необходимое для функционирования веб-сайта [2].

Веб-сервис – это реализация абсолютно четких интерфейсов обмена данными между различными приложениями, которые написаны не только на разных языках, но и распределены на разных узлах сети [3].

Геоид – сложная фигура планеты Земля, ограниченная уровенной поверхностью моря [4].

Геокодирование – это процесс преобразования описания местоположения (например, координат, адреса или названия места) в местоположение на поверхности Земли [5].

ГИС-вьюер – один из видов ГИС, предоставляющий доступ к картографическим данным преимущественно на чтение.

ГИС-сервер – сервер, обеспечивающий предоставление удаленного доступа к пространственным и картографическим данным, находящимся в базе данных посредством сетевых запросов.

ГИС-сервис – веб-сервис, обеспечивающий доступ к пространственным данным, полученным от ГИС-сервера, их обработку, анализ, поиск и визуализацию [6].

Интерфейс – совокупность средств и правил, обеспечивающих взаимодействие вычислительных систем, входящих в их состав устройств, программ, а также пользователя с системой; последний из них носит особое название интерфейса пользователя, в современных программных средствах оформляется графически [7].

Контент – вся информация, содержащаяся на веб-сайте.

Мобильная ГИС – класс сетевых геоинформационных систем, предоставляющих универсальный комбинированный доступ к удаленным геоинформационным инструментам, приложениям и базам данных, используемым пользователем независимо от того, в каком географическом месте он находится в определенный момент времени [7].

Пространственный анализ – группа функций, обеспечивающих анализ размещения, связей или иных пространственных отношений пространственных объектов [7].

Пространственный объект – такой объект, который хранит свое географическое представление, представленное обычно в виде точки, линии или полигона, в качестве одного из свойств (полей) в строке [8].

СУБД – комплекс программ и языковых средств, предназначенных для создания, ведения и использования баз данных [7].

Тайл (англ. tile) – фрагмент растровой карты в виде квадрата. При загрузке карта разбивается на тайлы.

UX/UI дизайн – это проектирование любых пользовательских интерфейсов в которых удобство использования так же важно как и внешний вид [9].

1.2 Классификация веб-сервисов. ГИС-сервисы

Веб-ГИС является разновидностью геоинформационной системы, базирующейся на веб-технологиях доступа к данным. Использование веб-технологий позволяет не привязываться к конкретному рабочему месту и предоставлять доступ к данным и функционалу системы через интернет прямо в окне браузера. При этом функционал веб-ГИС практически не уступает настольной геоинформационной системе: пользователь может добавлять, редактировать и анализировать данные, осуществлять поиск необходимой ему информации, использовать карты, назначать географические идентификаторы объектов и многое другое [10]. Любое веб-приложение, будь то веб-ГИС или обычный веб-сайт, строится из веб-сервисов. Можно выделить два типа веб-сервисов, которые могут служить основой для построения прикладных геоинформационных систем:

— служебные сервисы. Модульная архитектура разработки региональной геоинформационной системы и позиционирование ее компонент как составляющих для других веб-приложений сформировала необходимость строгой формальной спецификации информационного обмена между компонентами;

— публичные сервисы. В отличие от упомянутых выше служебных сервисов – эти сервисы являются публично доступными, технологически они практически не отличаются от служебных, но нарушить работоспособность системы они не могут.

При разработке веб-ГИС используются преимущественного ГИС-сервисы. Отличие веб-сервиса от ГИС-сервиса состоит в том, что ГИС-сервис предназначен для работы с пространственными данными. В основе каждого ГИС-сервиса лежит геоинформационный ресурс: карта, база данных, инструмент геообработки и др. ГИС-сервис является связующим звеном между исходным ресурсом и клиентским приложением (настольной или мобильной ГИС, веб-приложением и др.), выполняя передачу данных от ресурса к клиенту и обратно, обслуживая различные запросы пользователей. Клиентскому приложению, чтобы обратиться к геоинформационному ресурсу, не нужно понимать форматы хранения географических данных и функций – все это берет на себя ГИС-сервис. Достаточно послать стандартный запрос к ГИС-сервису по сети Интернет, чтобы получить нужный результат.

ГИС-сервисы могут использоваться следующими способами:

— как часть веб-ГИС – для реализации на веб-сайтах возможностей доступа к пространственным данным и функциям геообработки;

— как независимые веб-сервисы – для предоставления другим организациям возможности встраивания сервисов в собственные приложения на платной или бесплатной основе.

1.3 Примеры применения ГИС-технологий в веб-проектах

Существует множество отраслей, в которых целесообразно представление информации посредством применения геоинформационных технологий в веб-проектах и веб-системах, образуя в совокупности веб-ГИС. С помощью этих технологий реализуется отображение объектов на карте, инструменты пространственного анализа и т.д. Обычно, в таких отраслях основная часть данных – пространственные, а их удобное визуальное отображение и набор инструментов для работы с ними позволяют получать определенные выгоды. Среди таких выгод можно отметить следующие:

— обеспечение сотрудников компании удобным интерфейсом для работы с пространственными данными;

— понятная визуализация пространственных данных позволяет привлекать больше посетителей и вызвать больший интерес к компании;

— возможность отображать пространственные данные по правилам картографической семиотики и картографической топонимики;

— стандартизация использования геоинформационных ресурсов в организации (пользователи работают с одним и тем же представлением данных и функций);

— для сторонней организации, использующей ГИС-сервис – возможность доступа к актуальным данным другой предметной области без необходимости самостоятельно разрабатывать, обновлять и поддерживать эти данные.

Реализация веб-ГИС возможна практически во всех отраслях, например, медицина, строительство, природоохрана.

Примером использования веб-ГИС технологий в природоохранных учреждениях является веб-ГИС «Фаунистика». Система позволяет пользователям вносить и хранить:

— точечную информацию в основной базе данных по видам птиц (БД наблюдений);

— точечные данные, импортированные в систему в формате GPX;

— географические слои, состоящие из полигональных, линейных и точечных объектов, импортированных в систему в формате kml/kmz (данные форматы описаны в разделе 2.2), либо созданных вручную на карте системы.

Веб-ГИС «Фаунистика» за 2 года своего существования стала эффективным механизмом сбора информации по многим редким видам птиц, а также полноценной системой для ведения кадастра этих видов, осуществления мониторинга и охраны мест их обитания с применением возможностей ГИС [11]. На рисунке 1 представлен интерфейс Веб-ГИС «Фаунистика».

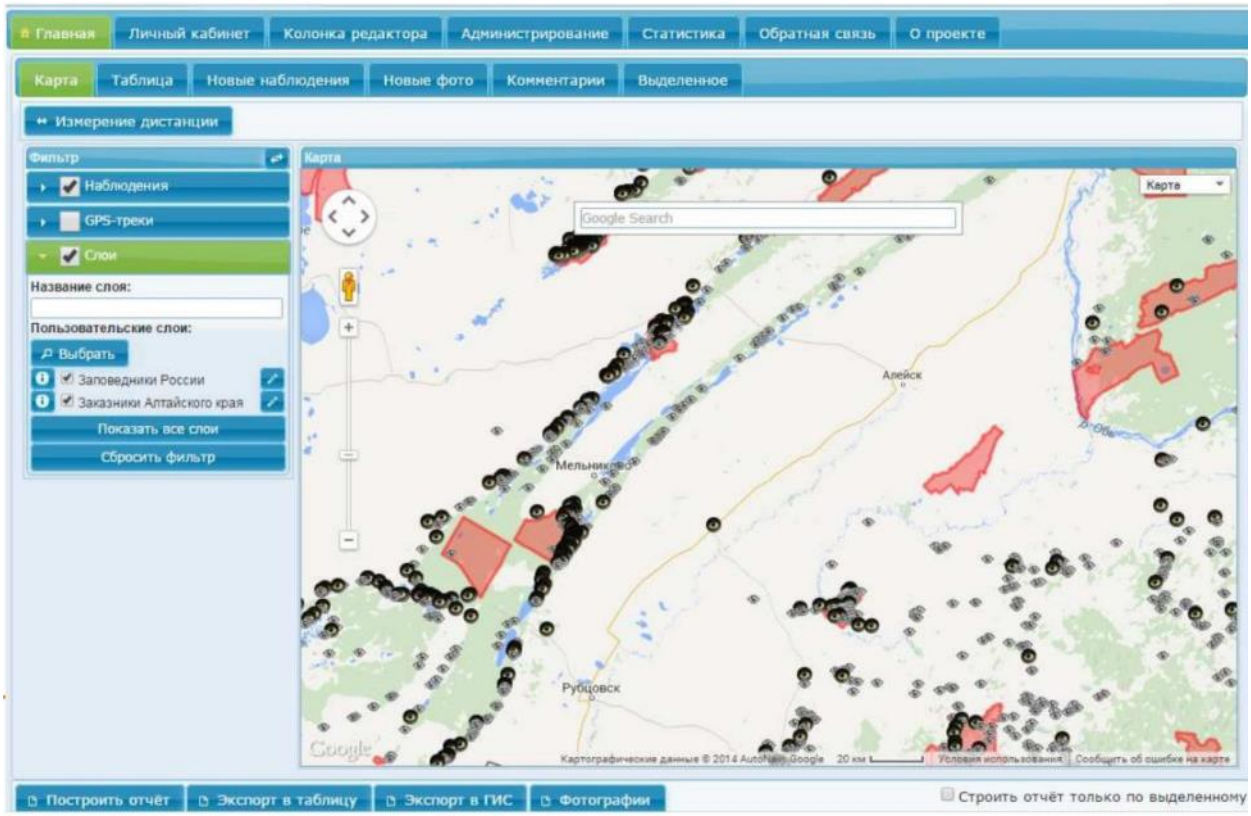

Рисунок 1 – Веб-ГИС «Фаунистика»

В строительной отрасли примером применения интернет-картографических технологий является ГИС для оперативного контроля строительства волоконно-оптических линий связи (ВОЛС). В системе реализовано взаимодействие между настольной ГИС и веб-компонентом для отображения пространственных данных. Данная веб-ГИС позволяет сортировать данные, редактировать данные, настраивать стиль отображения данных [12]. На рисунке 2 представлен интерфейс Веб-компонент ГИС для оперативного контроля строительства ВОЛС.

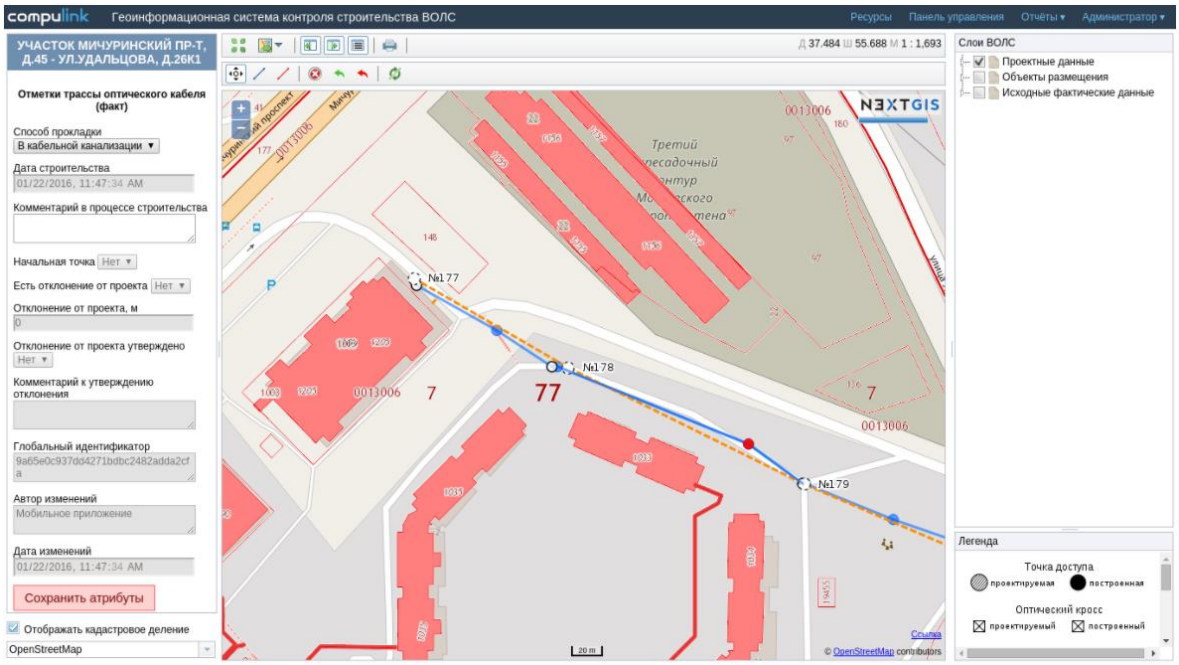

Рисунок 2 – Веб-компонент ГИС для оперативного контроля строительства ВОЛС

Веб-компонент данной ГИС также адаптирован под мобильные устройства. Это позволяет не привязываться к конкретному типу устройств для доступа к системе (например, настольному компьютеру). При такой реализации часть инструментария ГИС может быть скрыта, но при этом другая часть может быть добавлена. Например, современные мобильные телефоны оснащены GPS-приемником, позволяющим определять точное местоположение устройства, записывать маршруты и многое другое. Эти и другие возможности могут расширить функционал ГИС в мобильном виде.

Существует также отдельная категория веб-ГИС, именуемая ГИС-вьюерами. Одним из представителей ГИС-вьюеров является система 2ГИС. На рисунке 3 представлен интерфейс ГИС-вьюера 2ГИС.

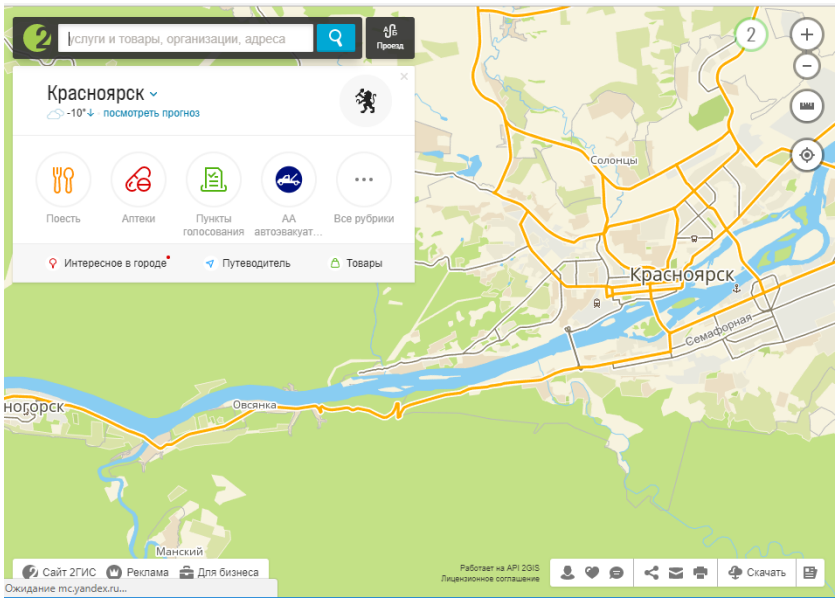

Рисунок 3 – ГИС-вьюер 2ГИС

Основная особенность таких систем – предоставлять пользователю преимущественно информацию справочного характера с доступом на чтение. Таким образом, используя 2ГИС пользователь может просматривать карту интересующей его местности, определять на ней своё местоположение, измерять расстояние между точками, осуществлять поиск интересующих мест, организаций и т.д. Также у 2ГИС имеется собственное мобильное приложение (рисунок 4).

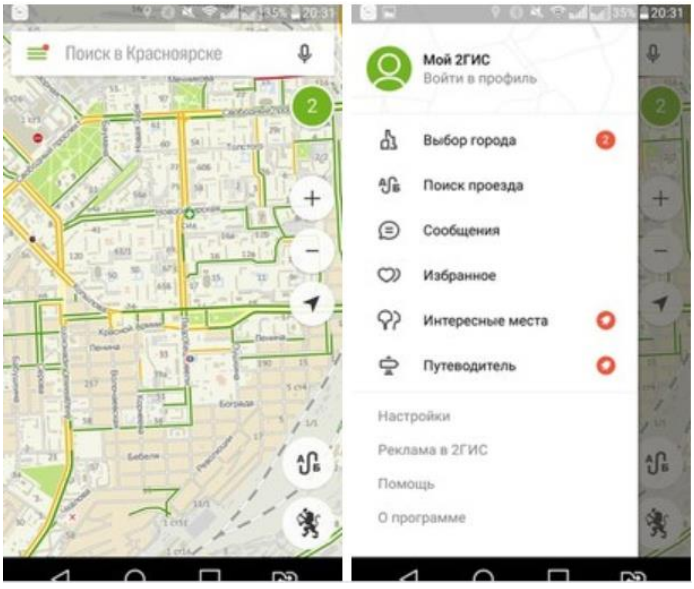

Рисунок 4 – Мобильное приложение 2ГИС

Из рисунка 4 видно, что приложение имеет адаптированный под экран устройства размер, его кнопки стали больше по сравнению с веб-версией (адаптация для пальчиковой навигации). Но в то же время можно заметить, что основной инструментарий приложения располагается в правой части приложения, тогда как главное меню – в левой. Это позволяет сделать выводы о том, что при размере экрана мобильного устройства более ~4,5 дюйм, использование данного приложения одной рукой становится затруднительным.

1.4 Платформы для веб- и мобильных ГИС

Во времена начала широкого распространения Интернет-технологий единственным способом получения доступа к пространственным данным, опубликованным в сети Интернет, служил компьютер с установленным на нем интернет-браузером. Это мог быть Internet Explorer, Opera, Google Chrome, Mozilla Firefox и т.д. Затем, примерно с 2008 года, в регионах России получил широкое распространение стандарт третьего поколения мобильной связи 3G. С тех пор мобильные устройства стали постепенно занимать нишу платформ, обеспечивающих доступ в сеть Интернет, что и повлияло на производителей веб-приложений, заставляя их все больше и больше внимания уделять адаптации своих продуктов под экраны с меньшим разрешением.

Примерно в то же время начался выпуск флагманских мобильных телефонов, работающих под операционной системой Android с открытым исходным кодом. Это событие стало стартовой точкой для разработки так называемых «нативных» приложений. Это такие приложения, код которых компилируется в машинные инструкции и выполняется непосредственно процессором устройства. Несмотря на то, что вслед за этим производители настольных браузеров стали выпускать мобильные версии своих продуктов, установленные непосредственно на мобильную платформу приложения, стали более удобным способом работы с данными. С такими приложениями работать удобнее, и они позволяли реализовать гораздо больший функционал с учетом

особенностей технологий мобильных устройств (например, наличие акселерометра и гироскопа, а также навигатора), нежели чем с обычным веб-приложением, просто адаптированным под размер экрана устройства. К тому же, мобильные приложения больше адаптированы под выполнение различных задач, в то время как веб-страницы по большей своей части дают доступ к информации.

Существует также отдельная категория устройств – мобильные планшеты. В зависимости от установленной на них операционной системы, их можно отнести либо к мобильным устройствам, либо к персональным компьютерам.

1.5 Минимальные требования к персональному компьютеру для разработки ГИС-приложений

В данном разделе приведены рекомендации по комплектации устройства для разработки веб и мобильных ГИС.

Произведение вычислений на персональном компьютере (далее – ПК) во время разработки приложения является одним из процессов, сильно повышающих загруженность компьютера. Комплектующие ПК для разработки должны уметь справляться с высокой нагрузкой при компиляции и интерпретации программного кода, либо при исполнении приложения.

Одной из технологий повышения производительности ПК является технология распараллеливания процессов. Согласно работе [13], в которой рассмотрены способы распараллеливания вычислений и закон Амдала, ускорение выполнения программы за счёт распараллеливания её инструкций на множестве вычислителей ограничено временем, необходимым для выполнения её последовательных инструкций. Из [14] следует, что сочетание мощных CPU с GPU позволит решать и задачи с большим количеством последовательных кодов, удовлетворяя требованиям закона Амдала и обрабатывая с высокой скоростью коды, которые могут быть распараллелены. При этом следует определиться, какая сторона приложения прогнозируется быть более ресурсоемкой –

графическая или вычислительная. При вычислительной ресурсоемкости допустимо совмещать связку менее мощный GPU – мощный CPU, и наоборот.

Как правило при тестировании производительности платформ акцент делается на процессорозависимые приложения. Но скорость системы зависит не только от центрального процессора. Степень зависимости программы от параметров памяти (оперативного запоминающего устройства – далее ОЗУ) во многом определяется характеристиками центрального процессора – прежде всего, размером его кэша, так как при увеличении объема кэш-памяти рабочая область программы (наиболее часто используемые данные) может поместиться целиком в кэш процессора, что качественно ускорит программу и сделает её малочувствительной к характеристикам памяти. Кроме того, важно, как часто в коде программы встречаются сами инструкции обращения к памяти. Если значительная часть вычислений происходит с регистрами, велик процент арифметических операций, то влияние скорости памяти снижается. Тем более что современные ЦП умеют изменять порядок выполнения инструкций и начинают загружать данные из памяти задолго до того, как те реально понадобятся для вычислений [15]. Следовательно, при наличии подходящей по мощности связки CPU + GPU можно не получить высокой производительности ПК, имея низкий объем ОЗУ.

2 Способы представления данных в ГИС

Во время работы веб- и мобильных ГИС производятся манипуляции с геоданными – отображение, различные вычисления. Существует две основные модели геоданных в ГИС – растровая и векторная. Растровая модель данных – цифровое представление непрерывных последовательностей реального мира в виде набора дискретных двухмерных объектов – пикселов (аналогия – представление изображений в компьютерной графике). Векторная модель данных – цифровое представление дискретных пространственных объектов реального мира в виде набора дискретных объектов – точек, линий, полигонов. Эта информация образует класс координатных данных ГИС.

Векторная модель данных обладает следующими преимуществами [16; 17]: высокая географическая точность, данные могут быть представлены в исходном разрешении и форме без обобщения, эффективное выстраивание топологии. Недостатки векторной модели данных: целостные данные плохо хранятся и отображаются в виде векторов, обязательное условие преобразования в топологическую структуру, а также алгоритмы векторной обработки с множеством функций сложны и оказывают большую нагрузку на ПК.

Растровая модель данных обладает следующими преимуществами [16; 18]: идеально подходит для математического моделирования и количественного анализа, отображает непрерывно охватываемую территорию, растровые данные проще для обработки и обеспечивают более высокое быстродействие, ввод растровых данных менее трудоемкий для системы ПК. Недостатки растровой модели данных: географические объекты характеризуются менее точной информацией о местоположении и размерах, требуют больших объемов памяти, обработка данных связанных атрибутов может быть громоздкой при наличии больших объемов данных.

Обычно при создании веб-ГИС и мобильных ГИС-приложений оперируют векторными данными. Они предоставляют информацию о том, где расположен тот или иной объект и обо всех атрибутах объекта.

Точечные, линейные и полигональные объекты хорошо подходят для отображения некоторых объектов ландшафта, например, деревьев, дорог и зданий. Другие объекты отобразить при помощи векторных объектов сложнее. Например, поля состоят из множества участков с разным цветом и плотностью покрытия.

Решением этих проблем является использование растровых данных. Часто в мобильной и веб-разработке используются растровые данные в качестве подложки под векторные слои, чтобы улучшить восприятие содержащейся в них информации. Человеческий глаз очень хорошо распознает образы, поэтому использование изображения под векторными данными делает карты более понятными и удобочитаемыми. Растровые данные хорошо подходят не только для изображений реальной поверхности (например, спутниковые изображения или аэрофотосъемка), но и для отображения абстрактной информации [19].

Для привязки необходимо правильно выбрать проекцию в зависимости от некоторых условий, описанных в разделе 2.1 данной работы.

2.1 Правила выбора проекции

Выбирая для карты ту или иную проекцию, в расчет принимается ряд обстоятельств, и в первую очередь назначение карты, из которого выбор проекции осуществляется с минимизированием искажений в пользу главной решаемой задачи проекта. Искажение (деформация) изображения, выражающееся в изменениях масштаба длин, присуще всем картографическим проекциям, – это их основное свойство. Но проекции различаются по характеру искажений (равноугольные, равновеликие, равнопромежуточные произвольные), по величине искажений и их распределению. Вообще говоря, проекции могут иметь отдельные точки, линии или даже систему линии, где сохраняется главный масштаб. В азимутальных проекциях – это точка касания плоскости, в конических – параллель касания конуса (или параллели сечения) и т. п. Такие точки и линии называются точками и линиями нулевых искажений.

Искажения возрастают по мере удаления от точек или линий нулевых искажении. Другими словами, они возрастают с увеличением размеров картографируемой территории. Минимизация искажений необходима для верного анализа содержимого карты (как визуального, так и расчетного) [20].

К примеру, цилиндрические проекции, чрезмерно искажающие площади фигур в высоких широтах, оказываются наиболее удобной для мореплавателей. Но такие проекции недопустимы на тех картах, где желательно сохранить правильное соотношение площадей, например, на картах политических, экономических, картах, показывающих размещение населения, сельского хозяйства и т. д. Если карта будет служить для измерения площадей, предпочтителен выбор равновеликой проекции. К произвольным проекциям прибегают в тех случаях, когда нежелательны чрезмерные искажения площадей и углов. Но мало сказать, что проекция для данной карты должна быть равновеликой или равноугольной, так как тех и других существует довольно много. Их выбор обусловлен размерами, формой и положением изображаемой на карте территории.

Цилиндрические проекции удобны для экваториальных стран, в конических очень хорошо располагаются страны, вытянутые вдоль параллели, независимо от того, на каких широтах, близких или далеких от экватора, эти страны находятся. Для территорий округлых очертаний применяют проекцию Бонна или одну из перспективных. Для стран, вытянутых вдоль меридиана, можно использовать цилиндрические проекции, но при другом положении оси цилиндра. Наконец, при прочих равных условиях предпочтение отдается тем проекциям, которые проще для вычисления и построения [21].

Зачастую в мобильной и веб-разработке ГИС-приложений используются карты-подложки от компаний Google Maps, Яндекс.Карты, Open Street Map, Bing и т.д. По умолчанию эти подложки имеют проекцию EPSG:3857 – WGS84 Web Mercator. Обычно сервисы используют сферическую проекцию Меркатора.

Если набор первоначальных данных не соответствует желаемой проекции, целесообразно воспользоваться библиотекой PROJ4, написанной на языке

Python. Библиотека поддерживает множество разных типов проекций, вместе с ней устанавливается обширная библиотека предопределенных проекций, но если предустановленных проекций недостаточно, то возможно определение и пользовательских проекций. Библиотека позволяет производить перепроецирование между системами координат, а также проводить вычисления на геоиде [22].

2.2 Форматы хранения пространственных данных

Форматами растровой модели данных являются распространенные графические форматы:

— TIFF – поддерживает сжатие как с потерями, так и без потерь. После привязки растровые данные поставляются в формате GEOTIFF;

— GIF – способен хранить сжатые данные без потери качества в формате не более 256 цветов;

— JPEG – поддерживает сжатие как с потерями, так и без потерь;

— PNG – спроектирован для замены устаревшего и более простого формата GIF, а также для замены значительно более сложного формата TIFF.

Основными векторными форматами являются:

— SHP – позволяет хранить следующие различные типы геометрических объектов: точки (мультиточки), линии (ломаные), многоугольники и другие объекты. Описывается несколькими файлами записей с определенными расширениями: .shp – позиционные данные, .shx – индекс формы пространственных данных, .dbf – атрибутивные данные, .prj – файл с описанием проекции;

— GML – словарь XML определенный Open Geospatial Consortium (OGC) для описания географических данных;

— KML/KMZ – язык разметки на основе XML для представления трёхмерных геопространственных данных в программе «Google Планета Земля»;

— DGN, DWG, DXF – специализированные векторные форматы. Являются производными форматами программ, работающих с векторной графикой;

— GeoJSON – открытый формат, предназначенный для хранения географических структур данных, основан на JSON.

Атрибутивные данные можно предоставлять отдельно – например, загружать их по запросу пользователя или в процессе подготовки данных добавлять дополнительную информацию к имеющимся данным. Такие атрибутивные данные могут быть представлены в форматах XML и CSV.

Для работы с растровыми географическими форматами файлов данных существует библиотека GDAL. Как библиотека GDAL предоставляет вызывающему приложению единую обобщённую модель данных для всех поддерживаемых форматов файлов данных. Помимо этого, в состав GDAL входит набор вспомогательных программ, вызываемых из командной строки, для преобразования и обработки данных, например, для географической привязки растрового изображения.

2.3 Стандарты ГИС-сервисов Open Geospatial Consortium

Для стандартизации создаваемых ГИС-сервисов и обеспечения возможности доступа к ним из различных приложений часто придерживаются стандартов консорциума OGC (Open Geospatial Consortium). Open Geospatial Consortium (OGC). OGC был основан в 1994 с целью сделать географическую составной частью мировой информационной инфраструктуры. Члены OGC – поставщики технологий и их пользователи – совместно разрабатывают открытые стандарты интерфейсов и соответствующие стандарты кодирования [23].

Основными стандартами ГИС-сервисов являются:

— Web Map Service (WMS) – предоставляет простой интерфейс http-запросов для получения геопривязанных изображений карт нескольких распределённых пространственных баз данных. В ответе сервиса на запрос будет

одно или более изображений карты (в формате JPEG, PNG и др.), которые могут быть показаны в браузере или настольном приложении.

— Web Feature Service (WFS) – веб-сервис пространственных объектов, определяющий интерфейсы и операции, которые позволяют запрашивать и редактировать векторные пространственные данные, такие, как дороги или береговые линии;

— Web Coverage Service (WCS) – определяет интерфейсы и операции, позволяющие взаимодействовать с пространственными данными, называемыми “покрытие”. Этим термином описываются спутниковые снимки, результаты аэрофотосъёмки, цифровые модели рельефа и другие данные, представленные значениями в каждой измеряемой точке;

— Web Processing Service (WPS) – описывает правила для входящих и исходящих данных (запросов и ответов на них) для сервисов геопроцессинга (геообработки), таких, как пересечение полигонов и др. WPS определяет веб-интерфейсы доступа к сервису геопроцессинга. WPS может описывать как простые расчёты, такие, как вычитание одного набора пространственных данных из другого (например, разница между областями распространения гриппа по сезонам года), так и более сложные, такие, как глобальная модель изменения климата.

3 ГИС-серверы: обзор и тестирование производительности

Беспрецедентный рост различных картографических веб-сервисов и появление огромного множества пользовательских карт в сети Интернет дали достаточно широкое распространение Интернет-картографии. Наиболее широким способом распространения карт в сети Интернет является использование кэшированных тайловых изображений WMS (см. раздел 2.6). Использование кеширующих сервисов в составе ГИС-сервера значительно повышает его производительность и масштабируемость.

Размещение пространственных данных на ГИС-сервере обеспечивает:

— удаленный доступ к пространственным данным;

— обеспечение доступа к пространственным данным независимо от подключения к сети (при этом данные берутся из кэша сервера);

— получение данных согласно спецификации OGS (см. раздел 2.5);

— подготовка данных непосредственно на самом сервере перед передачей их клиенту.

В настоящее время существует большое разнообразие ГИС-серверов: MapServer, GeoServer, MapSurfer.NET, ArcGIS Server, QGIS Server, Mapnik/TileMill, Map Suite/Thinkgeo и многие другие. Все расмотренные ГИС-сервера поддерживают спецификации OGS, рассмотренные в разделе 2.6 и работу с форматами данных, описанными в разделе 2.5. В данной главе перечислены одни из самых популярных ГИС-серверов, произведен их сравнительный анализ между собой по пунктам производительности, удобности использования и т.д.

3.1 Обзор ГИС-серверов

ГИС-сервер состоит из диспетчера объектов сервера (SOM) и контейнеров объектов сервера (SOC). Диспетчер объектов сервера управляет набором сервисов, которые распределяются между одним или несколькими контейнерами

объектов сервера. Когда приложение делает прямое подключение к ГИС-серверу по локальной или глобальной сети, он устанавливает соединение с диспетчером объектов сервера. Схема системной архитектуры веб-ГИС показана на рисунке 5.

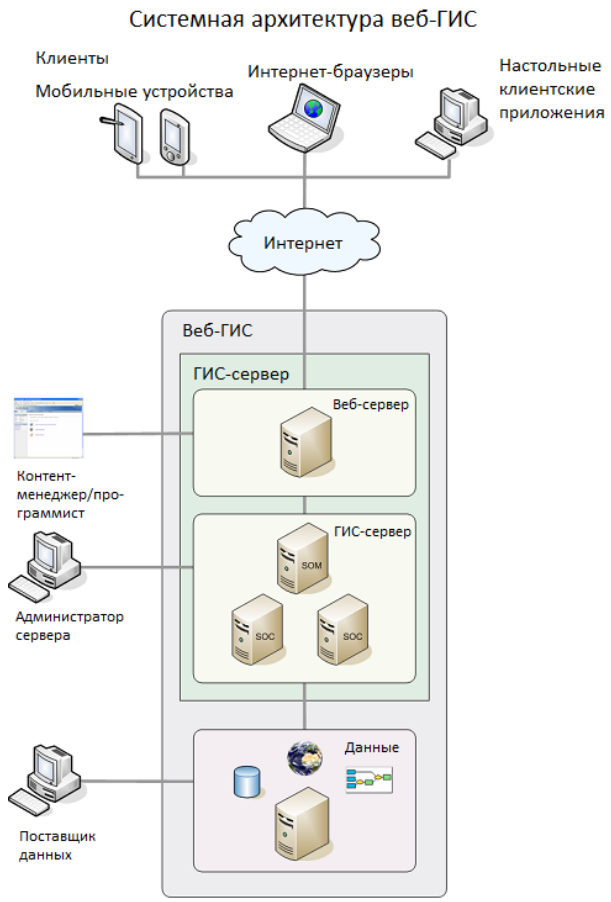

Рисунок 5 – Системная архитектура веб-ГИС

Контейнеры объектов сервера размещают службы, управляемые диспетчером объектов сервера. Все службы запускаются на всех контейнерных машинах, поэтому важно, чтобы все контейнерные машины имели доступ к ресурсам и данным, необходимым для запуска каждой службы. Можно установить значение емкости машины с контейнерами объектов сервера, чтобы ограничить количество запущенных служб, которые оно может одновременно размещать [24].

На сегодняшний день ГИС-сервер MapServer является одной из самых популярных сред создания ГИС-сервисов с открытым кодом. ГИС-сервер MapServer разрабатывался Университетом Миннесоты совместно с Департаментом Природных Ресурсов Штата Миннесота и NASA, а в настоящее время поддерживается как один из проектов ассоциации OSGeo. ГИС-сервер MapServer обладает возможностью работы практически на широко распространенных платформах (в том числе Windows, Linux, Mac OS, Solaris), большим количеством функциональных возможностей, легкостью интеграции с различными СУБД и открытостью кодов. ГИС-сервер MapServer позиционируется не как конечное приложение, а как среда разработки.

ГИС-сервер MapServer является очень мощным инструментом создания картографических веб-сервисов и по своей функциональности не уступает платному ПО, а по части легкости переконфигурирования и интеграции с СУБД превосходит многие из них.

К основным достоинства программы можно отнести следующие:

— возможность работы на практически любых платформах;

— поддержка большого числа растровых и векторных форматов данных;

— полное соответствие стандартам Open GIS Consortium (см. раздел 2.6);

— возможность переконфигурирования и программирования с использованием языков программирования Perl, PHP, Java, C, Python и др.

Хотя ГИС-сервер MapServer отлично работает со статическими источниками данных, его настройка представляет собой длительный ручной процесс [25].

ГИС-сервер GeoServer – это приложение, написанное на языке Java, обслуживающее карты и данные для их отображения в сторонних клиентах. Также, как и MapServer, GeoServer является картографическим сервером с открытым исходным кодом, который реализует спецификации OGS (см. раздел 2.6). Однако, в отличие от MapServer, GeoServer реализует спецификацию WFS-T (WFS-Transaction). Это означает, что, используя GeoServer, можно не только получать данные для построения на их основе собственных карт, но также

редактировать полученные данные с последующим автоматическим обновлением исходной информации на сервере. Среди поддерживаемых форматов значатся все форматы, перечисленные в разделе 2.5. [26].

Другой интересной особенностью, отличающей GeoServer от MapServer, является поставляемая с GeoServer визуальная система управления файлами настроек и описания данных для проектов GeoServer. Эта система реализована в виде веб-интерфейса и предоставляет пользователю возможность интерактивного создания и изменения разрабатываемого картографического ресурса [26].

ГИС-сервер MapSurfer.NET – это бесплатная, современная платформа для получения картографических продуктов. ГИС-сервер MapSurfer.NET разработан для быстрой и гибкой работы как настольных, так и веб-приложений [27].

Основные характеристики ГИС-сервера MapSurfer.NET [27]:

— написан на C # под платформой .NET Framework;

— мощная архитектура плагинов MapSurfer.NET делает данный ГИС-сервер очень расширяемым. Это позволяет создавать новые плагины для различных компонентов системы;

— встроенный механизм маркировки. Набор сложных алгоритмов маркировки предназначен для создания надписей высокого качества на карте;

— не требует дополнительной настройки при установке. Сама установка производится посредством запуска конфигурации установки MapSurfer.NET;

— имеет собственное приложение MapSurfer.NET Studio – для редактирования стилей карт;

— имеет встроенные инструменты для кеширования тайлов.

ArcGIS Server представляет собой программное обеспечение, в котором основной идеей является предоставление географической информации внутри организации, а также для любого интернет-пользователя. Это осуществляется с помощью веб-сервисов, позволяющих мощному серверному компьютеру получать и обрабатывать информационные запросы, отправляемые другими устройствами. ArcGIS Server делает веб-ГИС доступной на планшетных ПК,

смартфонах, ноутбуках, настольных рабочих станциях и любых других устройствах, которые могут подключиться к веб-сервисам [28].

Для начала работы с ArcGIS Server необходимо подготовить аппаратное и программное обеспечение, данные, а затем настроить веб-сервисы ГИС [28].

Аппаратное обеспечение, используемое для сервера, должно быть гораздо мощнее обычных настольных компьютеров. Для ArcGIS Server требуется машина, на которой можно установить 64-разрядную операционную систему. Архитектура ArcGIS Server является масштабируемой, т.е. при необходимости увеличения вычислительной мощности можно добавить дополнительные машины. Данным сервером поддерживаются запрос, извлечение и репликация данных базы геоданных, геообработка, геокодирование данных [28]. Стоимость приобретения ArcGIS Server для коммерческого использования на 22.05.2018 г. составляет 1 301 040 руб.

ГИС-сервер QGIS Server реализует WMS-сервис, используя те же библиотеки, что и настольное приложение Quantum GIS (QGIS). Карты и шаблоны для печати, созданные в настольной ГИС QGIS, могут быть опубликованы в Интернете простым копированием файла проекта в директорию сервера. В результате веб-карты выглядят так же, как и в настольном приложении. QGIS Server обычно работает как CGI/FastCGI модуль веб-сервера Apache. Основные характеристики QGIS Mapserver [29]:

— открытый WMS-сервер;

— представляет собой FastCGI/CGI приложение, написанное на C++;

— управление символикой из настольного клиента QGIS;

— встроенный генератор PDF;

— создание карт в QGIS в стиле WYSIWYG;

— расширенная картографическая символика.

ГИС-сервер Mapnik/TileMill – это ГИС-сервер, поставляемый в виде открытой библиотеки для рендеринга растровых карт. ГИС-сервер Mapnik/TileMill был разработан специально для проекта OpenStreetMap российским программистом Артемом Павленко, и сегодня используется в том

числе и другими ресурсами. Основная заявленная цель Mapnik/TileMill – получение качественно отрисованных карт.

ГИС-сервер Mapnik/TileMill предоставляет Python API, что делает возможным вызов функций Mapnik/TileMill (написанного на C++) из Python. Реализовано это в виде так называемых байндингов – специальных обёрток, позволяющих использовать возможности библиотек, написанных на одном языке программирования в приложениях, написанных с использованием другого (для Mapnik/TileMill существуют байндинги для таких языков программирования как JavaScript, Ruby и Java) [30].

Основные характеристики Mapnik/TileMill [30]:

— лицензия: LGPL (библиотеки под этой лицензией разрешается использовать для создания программ под другими лицензиями путем компоновки);

— поддерживаемые платформы: UNIX, Windows;

— поддерживаемые языки программирования: C++, Python;

— поддерживаемые графические элементы: точка, линия с заливкой и текстурой, полигон с заливкой, текстурой, экструзией (псевдо-3D), надпись к точке, линии, полигону, щиток (картинка и надпись, например, для обозначения трасс), векторный указатель (например, стрелка вдоль улицы).

ГИС-сервер Map Suite – это набор проприетарных инструментов ГИС и серверных API, созданный компанией ThinkGeo, предназначенный для развертывания в частных сетях, встроенных приложениях или размещенных на облаке ThinkGeo [31].

Основные особенности Map Suite [31]:

— инструменты для реализации маршрутизации и геокодирования;

— возможно использование на полатформах с операционными системами Windows, Linux, macOS;

— архитектурный стиль взаимодействия компонентов распределённого приложения в сети REST. Доступ к облачному серверу ThinkGeo можно получить с любого клиента, поддерживающего REST;

— нет инфраструктуры для поддержания. Поддерживаются серверы, балансировщики нагрузки, обновления программного обеспечения и т. д. Серверы Map Suite постоянно обновляются до новейших наборов данных продукта;

— масштабируемый и избыточный;

— гибкое использование транзакций и отчетность. Дополнительно предоставляется API для мониторинга использования и получения информации о том, как используются транзакции.

3.2 Сравнительный анализ ГИС-серверов

При проведении сравнительного анализа ГИС-серверов основное внимание уделяется сопоставлению производительности различных существующих ГИС-приложений и библиотек в генерации карт, использующих генерацию тайлов. Целью сравнения является обеспечить сопоставление производительности между различными бесплатными, открытыми и коммерческими приложениями и инструментами для рендеринга карт. Как правило, производительность таких программ зависит от различных факторов, таких как выборка данных, рендеринг функций, квантование изображений, запись на диск, перераспределение памяти и т.д. Все эти факторы влияют на производительность рендеринга тайлов в целом.

Приведенные [32] тесты ориентировочно показывают фактическую производительность тестируемых наборов инструментов, так как их рабочие характеристики зависят от платформы, поэтому исследования были проведены на машине с операционной системой Windows. Кроме того, довольно сложно правильно настроить набор инструментов, поскольку они имеют разнообразный набор параметров.

Эксперименты были выполнены на компьютере с процессором Intel Core i5-2500 с частотой 3,30 ГГц под управлением Windows 7 Professional x64 с установленным ОЗУ 8 ГБ. Среды выполнения подготовленного тестового приложения JRE 7/8 (x64) и .NET Framework 4.5 (x64). Тесты проведены на наборах данных, полученных от OpenStreetMap, а именно с использованием shape-файлов наземных зон (континентов и островов), взятых веб-сайта OpenStreetMapData. Использовались два shape-файла (разделенные версии) с данными в проекциях WGS84 и Меркатора с размером 400 Мб.

В таблице 1 приведены данные о тестируемых программных пакетах ГИС-серверов.

Таблица 1 – Данные о программных пакетах тестируемых ГИС-серверов

ГИС-сервер	Программный язык	Среда запуска	Архитектура	Версия	Дата выхода
GeoServer	Java	JRE 8/9	x64	2.7.0	20.03.2015
MapServer	C/C++, C#	Средства ОС	x86	6.4.1	02.01.2014
Map Suite	C#	.NET 4.0	x64	8.0.0.0	22.12.2014
MapSurfer.NET	C#/C++	.NET 4.5	x64	2.0.3	02.06.2015
Mapnik/TileMill	C++/Node.js	Средства ОС	x86	2.2/0.10.1	10.10.2012

Как видно, библиотеки и инструментальные средства написаны на разных языках программирования, таких как C/C ++, C # и Java. Все тестируемые ГИС-серверы поддерживают индексирование shape-файлов. При этом TileMill и MapSurfer.NET используют хранилище mbtiles, в отличие от остальных ГИС-серверов, использующих каталоги на жестком диске компьютера. В каждом тестировании ГИС-серверы запускались два раза и учитывался лучший показатель времени запуска. Этапы тестирования представлены в приложении А.

В результате тестирования ГИС-сервер MapServer показал очень высокую скорость отрисовки тайлов для уровней масштабирования z0-z7. При большем масштабировании (z8-z9) ГИС-сервер MapServer был преодолен двумя другими ГИС-серверами, такими как Mapnik/TileMill и MapSurfer.NET. В целом, Mapnik/TileMill продемонстрировал очень хорошую производительность практически во всех тестах. Показатель производительности в 1,85 раза ниже, чем у ГИС-сервера MapServer. Вероятно, результатом этого является известная проблема в Mapnik при использовании перепроецирования с использованием библиотеки proj4.

ГИС-сервер MapSurfer.NET имеет средние показатели. Он не продемонстрировал каких-либо выдающихся результатов при отрисовке тайлов в небольших масштабах. Но при этом ГИС-сервер MapSurfer.NET продемонстрировал отличную производительность при рендеринге очень большого количества тайлов. Например, пропускная способность отрисовки тайлов достигла 1285 тайлов в секунду.

Кроме того, замечено, что как ГИС-сервер GeoServer, так и ГИС-сервер Map Suite требуют гораздо больше памяти для выполнения задач, чем другие тестируемые ГИС-серверы. Их потребление памяти составило около 700-770 МБ против 250-300 МБ другими.

В заключении можно отметить, что ГИС-сервер Mapnik/TileMill и ГИС-сервер MapSurfer.NET по результатам тестирования являются двумя самыми быстрыми ГИС-серверами для рендеринга тайлов. В тестах № 2 и № 3 эти инструментальные средства примерно на 400% и 700% быстрее, чем ГИС-серверы GeoServer и Map Suite.

4 ГИС-сервисы: обзор и сравнение функционала

Существует большое количество бесплатных программ с открытым исходным кодом, которые облегчают создание и настройку клиентов веб-сопоставления. В данной главе приведены характеристики и сравнение некоторых ГИС-сервисов. Все перечисленные ГИС-сервисы распространяются свободно.

4.1 Обзор ГИС-сервисов

ГИС-сервисы представляют собой готовый инструментарий, используя который пользователь может создавать собственные карты зачастую без установки какого-либо дополнительного программного обеспечения. Зачастую они играют значительную роль в построении сложных инфраструктур представления пространственных данных (SDI), позволяющих визуализировать пространственные данные из нескольких источников. Таким образом, ГИС-сервисы могут быть частью веб-приложений Geographic Information Systems (GIS), в которых пользователи могут напрямую взаимодействовать с услугами SDI, визуализировать, запрашивать и интегрировать их с локальными данными и инструментами ГИС.

Согласно [33] ГИС-сервисы могут быть разделены на следующие категории: библиотеки, обёртки библиотек, инструментарии, фреймворки, клиентские приложения.

Библиотеки – это готовый набор API, который позволяет создавать приложения на более высоком уровне программирования. Обычно предоставляемых ими функций, методов и классов достаточно для создания полноценной клиентской части веб-ГИС. Далее приведен список некоторых распространенных библиотек:

— JavaScript-библиотека OpenLayers – библиотека, написанная на языке Javascript, предназначенная для создания карт на основе программного интерфейса, подобного GoogleMap API или MSN VirtualEarth API;

— JavaScript-библиотека Leaflet – одна из наиболее популярных картографических JavaScript-библиотек, использующаяся на таких крупных сайтах, как Flickr, Foursquare, Data.gov, проектах Викимедиа, OpenStreetMap, MapBox, CloudMade, CartoDB и других.

Обёртки библиотек (англ. wrapper) являются промежуточным слоем между прикладной программой и библиотекой. Целью написания обёртки библиотеки может быть обеспечение работоспособности библиотеки в каком-либо (чаще скриптовом) языке, в котором прямой вызов функций этой библиотеки API затруднителен или невозможен. При использовании оберток библиотек разработка менее подвержена ошибкам, так как при кодировании потребуется писать гораздо меньше кода на языке программирования JavaScript. Перечень некоторых оберток библиотек:

— GWT-OpenLayers – это оболочка, написанная на языке программирования Java для библиотеки OpenLayers. Он позволяет проектам GWT использовать JavaScript API OpenLayers;

— OL4JSF представляет собой набор компонентов JSF для библиотеки OpenLayers, целью которого является повышение эффективности разработки карт на прикладном уровне;

— MapQuery – это плагин jQuery, упрощающий работу с библиотекой OpenLayers.

Инструментарии (англ. toolkits) – это наборы функций и методов, которые легко интегрируются в приложение. Инструментарий представляет собой более сосредоточенную библиотеку с определенной целью. Инструментарий чаще всего работает на более высоком уровне абстракции, чем библиотека, при этом часто состоит из набора библиотек. Более низкие уровни абстракции в инструментарии являются либо фиксированными, либо могут изменяться клиентским кодом. Далее приведен список некоторых инструментариев:

— ReadyMap SDK – инструментарий 3D-картографирования для WebGL/Javascript. ReadyMap SDK предлагает API-интерфейс для визуализации и взаимодействия с 3D-картами;

— GeoExt – инструментарий, созданный для разработки на языке JavaScript, который объединяет функциональность библиотеки OpenLayers с пользовательским интерфейсом библиотеки ExtJS.

Фреймворк – программный продукт, который служит для упрощения и унифицирования процесса разработки технически сложных или нагруженных проектов. Фреймворки предлагают комплексную и интегрированную платформу для создания приложений, используя общие функции, такие как управление памятью через платформу, многопоточные абстракции, динамические структуры (и общие структуры в целом). Список некоторых фреймворков:

— Mapbender – это фреймворк для управления контентом для служб геопространственных данных и картографических приложений. По словам создателей Mapbender [34];

— CartoWeb – фреймворк, написанный с использованием языка PHP5, является очень модульным и настраиваемым благодаря объектно-ориентированной архитектуре;

— MapFish – это гибкий и самодостаточный фреймворк для создания полноценных приложений для веб-картографии. MapFish основан на веб-инфраструктуре Pylons Python и расширяет ее геопространственной спецификой.

Клиентские приложения – конечные готовые приложения, созданные специально для разработки собственных веб-ГИС. Представляют собой веб- или настольные приложения, как правило содержащие графическую оболочку. Далее перечислены некоторые клиентские приложения:

— QGIS Web Client является клиентским приложением, базирующимся на OpenLayers и ExtJS. Предназначен для использования совместно с расширениями QGIS Server;

— Geoide – поставляется в виде веб-клиента, что позволяет настраивать веб-ГИС в окне браузера, подходит для использования на мобильных устройствах;

— msCross – это веб-клиент, основанный на технологии AJAX (WEB 2.0), первоначально разработанный как интерфейс Javascript для UMN Mapserver;

— Яндекс Конструктор карт – это веб-инструмент для создания схем проезда и нанесения объектов на карту.

4.2 Сравнительный анализ ГИС-сервисов

В данной главе приведено сравнение коммерческих и свободнораспространяемых ГИС-сервисов, а также сравнение всех выявленных в предыдущем разделе категорий ГИС-сервисов по их характеристикам.

4.2.1 Сравнение ГИС-сервисов, распространяемых на платной и бесплатной основе

ГИС-сервера поставляются на платной и бесплатной основе. Далее рассмотрены положительные и отрицательные стороны использования обоих видов ГИС-серверов согласно [35].

Положительные стороны использования платных ГИС-сервисов: удобный пользовательский интерфейс, широкий набор инструментов для пространственного анализа, полная документация, мощная техническая и пользовательская поддержка.

Отрицательные стороны использования платных ГИС-сервисов: проблемы масштабируемости, стоимость по количеству пользователей, выпускаются только для использования на компьютерах с установленной операционной системой Windows.

Положительные стороны использования ГИС-сервисов с открытым исходным кодом: хорошо подходят для разработки веб-ГИС с нестандартным

функционалом, легко масштабируемы, работают на большинстве операционных систем: Windows, MacOS, Linux и т.д., хорошо подходят для личного использования в виду отсутствия материальных затрат.

Отрицательные стороны использования ГИС-сервисов с открытым исходным кодом: высокий порог вхождения для использования, практически полное отсутствие технической и пользовательской поддержки, предоставляемая такими ГИС-сервисами документация зачастую бывает неполной, плохо описанной.

На рисунке 6 приведен график сравнения возможностей платных и свободнораспространяемых ГИС-сервисов.

Рисунок 6 – График сравнения платных и свободнораспространяемых ГИС-сервисов

Из графика видно, что коммерческие ГИС-сервисы (желтая линия) предпочтительны в том случае, когда необходимо иметь в разработке удобную среду с хорошей технической поддержкой. Это особенно актуально в больших компаниях, где разработкой и наполнением контентом веб-ГИС могут заниматься не только программисты-разработчики. С другой стороны, ГИС-

сервисы открытым исходным кодом (зеленая линия) проще настраивать, масштабировать и адаптировать к существующим программным системам. Хотя такие трудозатраты часто обходятся дорого.

4.2.2 Сравнение характеристик ГИС-сервисов

В приложении Д определена таблица, в которой приведено сравнение описанных в разделе 4.1 ГИС-сервисов по некоторым параметрам.

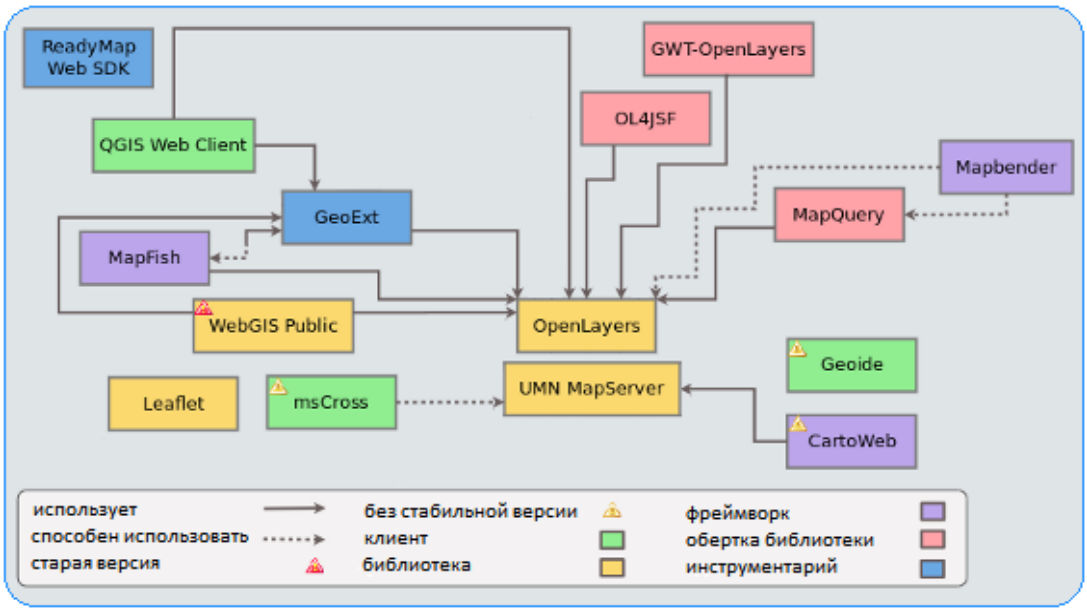

Рисунок 7 – Диаграмма взаимосвязей ГИС-сервисов

Опираясь на таблицу из приложения Д и рисунок 7 можно сделать вывод, что среди библиотек OpenLayers имеет поддержку OSGEO и также поддерживает передачу данных через OGS-протокол WFS, в отличие от Lealfet, но при этом занимает гораздо больше места на дисковом пространстве. Обёртка библиотеки OL4JSF имеет такие же характеристики, как и GWT-OpenLayers, но при этом занимает гораздо меньший объем памяти. Инструментарий GeoExt поддерживает OGS-протоколы WFS, WFS-T и имеет меньший размер, чем ReadyMap SDK. Фреймворк CartoWeb оказался самым легковесным среди других рассматриваемых фреймворков, но в то же время фреймворк Mapbender

может работать с большим количеством OGS-протоколов. Среди клиентских приложений больше всего протоколов поддержвает Geoide, меньше всего – QGIS Web Client.

5 Базы данных для хранения пространственных данных

Для хранения пространственных данных необходимо выбрать такую систему управления базами данных (СУБД), которая позволит не только хранить, редактировать и удалять данные, но и производить обработку геоданных.

5.1 Обзор систем управления базами данных

В данной главе приведен обзор систем управления баз данных (СУБД), рассмотрены их инструменты для работы с пространственными данными, а также приведены результаты тестирования с применением бенчмарка для сравнения СУБД Jackpine. Все рассмотренные базы данных способны хранить как векторные, так и растровые пространственные данные.

СУБД MySQL – это система управления реляционными базами данных, имеет условно платное распространение. MySQL осуществляет пространственные расширения по спецификации Open Geospatial Consortium (OGC). MySQL осуществляет подмножество типов SQL withGeometryTypes, среду, предложенную OGC. Этот термин относится к SQL-среде, которая была расширена с набором типов геометрии. Оцененный геометрией SQL столбец выполнен как столбец, который имеет тип геометрии. Спецификация описывает набор SQL-типов геометрии также, как функций на этих типах, чтобы создавать и анализировать значения геометрии [36].

СУБД MySQL еще не выполняет следующие свойства GIS [36]:

— дополнительные просмотры метаданных;

— спецификация OpenGIS предлагает несколько дополнительных просмотров метаданных. Например, просмотр системы GEOMETRY_COLUMNS содержит описание столбцов геометрии, одна строка для каждого столбца геометрии в базе данных;

— функция OpenGISLength() на LineString и MultiLineString в настоящее время должна быть вызвана в MySQL как GLength().

СУБД PostgreSQL – это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных. Данная СУБД имеет специальное расширение PostGIS, предназначенное для хранения в базе географических данных. PostGIS включает поддержку пространственных индексов R-Tree/GiST и функции обработки геоданных. PostGIS разрабатывается консалтинговой компанией RefractionsResearchInc как проект в области изучения технологий пространственных баз данных. Планируется сопровождение и дальнейшая разработка PostGIS в направлениях включения важной функциональности ГИС, полной поддержки OpenGIS, продвинутого топологического конструирования (покрытия, поверхности, сети), создания пользовательского интерфейса для просмотра и редактирования данных ГИС [37].

СУБД Oracle Database – это объектно-реляционная система поддерживающая некоторые технологии, реализующие объектно-ориентированный подход, то есть обеспечивающих управление создания и использования баз данных. Распространяется платно.

СУБД Oracle Spatial – это компонент базы данных, состоящий из типов данных, набора функций и процедур, а также внешних по отношению к базе данных веб-сервисов, которые позволяют эффективно хранить, быстро получать доступ и анализировать пространственные данные [38].

Преимущества хранения данных в Oracle Spatial [38]:

— индустриальный формат. Oracle Spatial поддерживается как стандартный формат хранения всеми крупными ГИС-вендорами и поставщиками пространственных данных, т.е. не требуется конвертировать данные в разные форматы;

— скорость работы. Обработка данных осуществляется в месте их хранения, что очень эффективно и масштабируемо. Операции обработки

эффективно распараллеливаются, чтобы использовать многопроцессорные системы;

— централизованное хранение. От этого выигрывает безопасность и надежность хранения.

СУБД Oracle Spatial – это де-факто стандарт хранения пространственных данных в корпоративном мире. Его понимают все основное ГИС ПО [38].

СУБД Informix® – это встраиваемая база данных, оптимизированная для данных OLTP и Internet of Things (IoT), разработанная компанией IBM. Informix обладает уникальной возможностью беспрепятственной интеграции SQL, NoSQL/JSON, управлением временными рядами и пространственными данными. Данная СУБД распространяется платно.

Пространственное решение СУБД Informix включает в себя ГИС на сервере базы данных Informix. Типы пространственных данных IBM Informix заключают в себе пространственные и не пространственные данные, обеспечивая беспрерывную точку доступа посредством языка SQL [39].

Типы пространственных данных СУБД Informix поддерживают спецификацию OGC, посредством обращений к СУБД на основе языка SQL3 для абстрактных типов данных (ADT). Эти типы данных могут хранить пространственные данные, такие как местоположение, ориентиры [39].

Для запуска SQL-запросов к пространственным данным используется DB-Access.

5.2 Тестирование СУБД

В данной главе приведен сравнительный анализ производительности СУБД MySQL 5.0.91, PostgreSQL 8.4.2, Informix 11.50. Анализ произведен на основании результатов исследования [40], где был разработан бенчмарк для сравнения СУБД Jackpine.

С ростом популярности веб-картографии поддержка СУБД работы с пространственными данными становится одной из важных частей СУБД.

Пространственные функциональные возможности в коммерческих и открытых СУБД сильно различаются, и для сравнения таких разнообразных предложений нет стандартного базового шаблона пространственной базы данных. Jackpine – это бенчмарк пространственной базы данных, разработанный в рамках текущих исследований по поддержке системного программного обеспечения для работы с пространственными базами данных. Бенчмарк включает в себя полный набор запросов, в которых используются пространственные функции, а также микро-бенчмарки, которые моделируют общие сценарии использования в реальном времени. Целью микро-бенчмарка является проверка способности вычислять основные топологические отношения и функции пространственного анализа.

Макро-бенчмарки, примененные в исследовании: геокодирование, поиск по координатам, вычисление дистанции между объектами их отношения друг к другу (пересечение, вхождение), последовательность четырех запросов с использованием пространственных и не пространственных объединений и агрегирующих операций, запросы по нахождению топологических взаимосвязей пересечения и равности, поиск объекта, который имеет отношение «является частью» к другому объекту.

Компьютер, на котором запускались тесты, оснащен процессором Intel Pentium 4 (2,4 ГГц) с ОЗУ 512 МБ и жестким диском объемом 240 ГБ. Операционная система – 32-битная Ubuntu 10.04 Lucid.

5.3 Результаты тестирования СУБД

Первый тест включает в себя время загрузки shape-файлов разного размера. Для загрузки shape-файлов в СУБД MySQL использовалась утилита GDAL ogr2ogr, для СУБД PostgreSQL – встроенная утилита shp2pgsql, а для СУБД Informix – встроенная утилита loadshp. Таблица, показывающая время, затраченное СУБД на импорт shape-файлов, индексирование и обновление состояния СУБД и результаты тестирования СУБД с применением микро- и макро-бенчмаркингов описаны в приложении Б.

В исследовании [40] авторы используют среднее геометрическое значение по всем запросам, где каждый результат времени нормализуется к эталонной СУБД. Поскольку только СУБД PostgreSQL поддерживает все запросы бенчмаркингов, решено использовать ее как базовую СУБД. В идеале оценка вычислена по всем запросам в тесте, исключая запросы, которые не поддерживаются MySQL и Informix. Вычисление происходит по формуле

$$Score_{DBMS_x} = \sqrt[N]{\prod_{q=1}^{N} \frac{Time_q^{DBMS_{ref}}}{Time_q^{DBMS_x}}},$$

где N – общее количество запросов к СУБД,

DBMS – эталонная СУБД PostgreSQL.

Результат вычислений представлен в таблице 2.

Таблица 2 – Результат проведения теста по бенчмарку Jackpine

Бенчмаркинг	PostgreSQL	MySQL	Informix
Микро-бенчмаркинг	1,00	2,21	5,84
Макро-бенчмаркинг	1,00	1,06	0,32

Хоть СУБД Informix и имеет гораздо более высокий балл по микро-бенчмаркингу, сочетание запросов, используемых в макро-бенчмаркинге, приводит к более низкому результату. СУБД MySQL показывает практически эталонный результат на макро-бенчмаркинге, хотя его преимущество на микро-бенчмаркинге практически в два раза. Однако, СУБД MySQL может возвращать много «ложноположительных» записей в результате, поскольку он не выполняет этап уточнения процесса выполнения пространственного запроса. В ходе тестирования было выявлено, что, количество записей, возвращаемых в ходе микро-бенчмаркинга, одинаково для PostgreSQL и Informix, тогда как MySQL возвращает большее количество записей.

6 Разработка мобильной ГИС

В данной главе описаны методики и технологии проектирования и конструирования мобильных приложений с использованием ГИС технологий.

Существуют специальные ГИС-сервисы, позволяющие реализовывать мобильные ГИС-приложения:

— eLite Map SDK – создание многофункциональных автономных мобильных приложений для устройств Android и iOS на базе отечественных технологий;

— технология CoGIS Mobile – создание на базе портала CoGIS мобильных геоинформационных приложений для неограниченного числа пользователей, позволяющих управлять контентом на мобильных устройствах путем синхронизации и обмена с корпоративными базами геоданных;

— платформа ArcGIS – быстрое создание «нативных» мобильных приложений на базе ArcGIS Runtime SDK, позволяющего использовать все возможности платформы ArcGIS в мобильных и встраиваемых устройствах: от простого отображения карты на экране до навигации и продвинутого пространственного анализа.

Далее приведен краткий обзор специализированных инструментов для создания собственного мобильного ГИС-приложения, описан процесс реализации некоторой функциональности, которая может быть использована в любом приложении, независимо от его сферы применения.

6.1 Исходный пакет инструментов для разработки мобильных ГИС

На данный момент существуют две самые распространенные мобильные операционные системы, под которые ведется разработка – iOS от компании Apple Inc., Android от компании Google Inc. и Windows Mobile от компании Microsoft.

На данный момент существует 3 вида мобильных приложений:

— веб-приложения. Иначе говоря, мобильные версии сайтов, запускаемые напрямую через браузер вашего устройства. Разработка ведется аналогично разработке веб-приложения для полной версии сайта, часто ограничен функционал;

— нативные приложения. Речь идет о приложениях, разработанных под «родную» платформу, то есть Android, iOS или Windows. Они загружаются напрямую из магазина приложений, оптимизированы с точки зрения взаимодействия с системой, расхода батареи и полноценного использования возможностей устройства;

— гибридные приложения. В данном пункте есть некоторое расхождение: существует точка зрения, по которой гибридные приложения являются веб-сайтами, разрабатываемыми по универсальной схеме для настольных компьютеров и мобильных устройств. Вторая версия, которая считается более универсальной, при этом не исключающей первую, гласит о том, что гибридные приложения – это некий компромисс между веб-приложениями и нативными, то есть загружаемые из магазина, имеющие оболочку, написанную на платформенном языке, но имеющие в той или иной степени веб-функционал [41].

Положительной стороной создания мобильных ГИС-приложений является возможность работы с пространственными данными непосредственно на местности, в отличие от настольных ГИС-систем. Каждый вид мобильных приложений реализуется конкретно определенным набором языков программирования.

6.1.1 Языковые средства программирования для разработки мобильных приложений

Самым простым в реализации видом мобильных приложений является мобильный вид обычного полнофункционального сайта. Для этого разработчику достаточно обладать знаниями языка разметки гипертекста HTML, построения

каскадных таблиц стилей CSS, а также языка программирования JavaScript. Чтобы адаптировать дизайн и функционал веб-приложения под мобильную версию, необходимо оперировать медиа-запросами CSS, а также средствами документно-ориентированной модели JavaScript DOM – все эти средства позволяют задавать определенное поведение веб-приложению при определенном размере экрана устройства. При создании мобильного веб-приложения разработчик не задумывается о специфике мобильных операционных систем, но при этом стоит учитывать особенности мобильных веб-браузеров.

Разрабатывая нативные приложения необходимо оперировать конкретным языком программирования в зависимости от операционной системы. Базовые языки для iOS – Objective C и Swift. Разработка нативных приложений под ОС Android ведется с помощью языка Java. Приложения для ОС Windows Mobile разрабатываются на языке C#. Существует также JavaScript фреймворк React Native для разработки кроссплатформенных приложений, позволяющий разрабатывать приложения одновременно под ОС iOS и ОС Android. При этом большая часть приложения имеет общий код для обеих ОС, а отдельные обработчики событий программируются для каждой ОС отдельно.

При разработке гибридных приложений могут использоваться любые языки программирования, в зависимости от того, какое приложение необходимо создать. В простейшем случае, для создания интерфейсной части может использоваться нативная часть (Swift, Java, C# и т. д.), а внутренние механизмы создаются, например, на HTML5 или JavaScript. То есть для того, чтобы перейти на другую платформу, придётся потратить меньшее количество времени, чем при создании стандартного нативного приложения. Минусы такого подхода – сниженное по сравнению с нативными приложениями быстродействие, меньше возможностей для взаимодействия с устройствами, больше вероятность появления ошибок.

Совмещение разработки мобильного приложения одновременно для всех основных мобильных ОС является сложной задачей, поскольку каждая ОС

использует различные подходы к проектированию UI/UX. При адаптации iOS приложения для работы с Android (и наоборот) необходимо учитывать как технические особенности разработки, так и традиционные пользовательские предпочтения.

6.1.2 Особенности UI/UX-проектирования приложений с элементами ГИС-технологий под мобильные операционные системы

Для создания единого стиля в интерфейсе приложений на iOS и Android можно воспользоваться дизайн-системами известных компаний. Дизайн-система – это целостный визуальный язык и его техническое отражение в виде библиотеки компонентов на едином репозитории, а также сопутствующих дизайнерских шаблонов [42]. Такой инструмент позволяет установить единую стартовую точку, на которую необходимо опираться, если появляются какие-либо вопросы касательно разработки.

Также для того, чтобы придерживаться единого дизайна всего приложения, вне зависимости от того, для какой ОС оно разрабатывается, подходит также спецификация от Google Material Design [43]. Одна из ключевых идей Material Design заключается в создании у пользователя интуитивного ощущения работы с реальными физическими объектами в рамках цифровой среды. По сути, это эмуляция трёхмерного пространства на плоскости экрана, но со всеми преимуществами, которые может дать виртуальная среда.

При разработке мобильных приложений с внедрением ГИС инструментов, взаимодействующих с картой (например, построение маршрутов и буферных зон, вычисление расстояний и т.д.) необходимо учитывать особенности проектирования интерфейса для экранов небольшого размера.

При выборе инструментария, который будет отображаться пользователю на главном экране, рекомендуется воспользоваться правилом «20-80». При таком подходе необходимо оставлять на главном экране 20% всей функциональности

приложения, на которую у пользователя будет тратиться 80% всего времени, проведенного в приложении.

Необходимо минимизировать количество кнопок и прочих элементов интерфейса на главном экране. Главным рабочим пространством в приложении является карта, соответственно она должна занимать в приложении ~90% всего экрана. Иконки должны быть размером не менее 42px – это минимальный размер кнопки, при котором будет удобно нажатие пальцем.

Так как ширина экрана мобильного телефона невелика, лучше всего производить какие-либо операции с объектами на карте, находящимися друг от друга на сравнительно небольшом расстоянии. К примеру, показывая построенный маршрут между двумя городами, будет невозможно охватить весь маршрут подробно при минимальном взаимодействии пользователя с приложением. Таким образом, чем больше расстояние между объектами, тем меньше подробность маршрута – это вынуждает пользователя просматривать его отдельные части, совершать больше действий в приложении. Если же объекты самодостаточны и приложение не предоставляет визуальных средств для их связывания между собой – расстояние между ними не будет иметь большого значения.

Интерфейс мобильного приложения должен быть отзывчив к повороту экрана. Это означает, что элементы приложения должны подстраиваться под поворот устройства, доступность интерфейса должна сохраняться независимо от горизонтального или вертикального положения устройства.

6.2 Инструменты пространственного анализа в мобильных ГИС

С помощью библиотек и фреймворков выбранного языка программирования разработчик имеет возможность создавать инструменты пространственного анализа в мобильном приложении с высокой производительностью и минимальными трудозатратами. Такие инструменты практически не уступают по точности вычисления аналогичным инструментам в

настольным ГИС [44]. Благодаря этому можно совершать различные вычисления пространственного характера, при необходимости редактируя элементы с последующим сохранением в базу данных, делать различные выборки. Перенос всей функциональности настольного ГИС-приложения в мобильное весьма труден, но обычно это нецелесообразно, так как производительности мобильного процессора не хватит на выполнение сложных операций наравне с персональным компьютером. Поэтому, мобильные ГИС-приложения обычно имеют весьма ограниченный функционал, не требующий больших производительных мощностей от мобильной платформы.

Стандартный набор инструментов, который должен присутствовать на каждой карте в мобильном приложении:

— инструменты масштабирования карты;

— показ текущего местоположения пользователя;

— получение координат местоположения пользователя;

— информационные окна при нажатии на интересующий объект на карте.

В мобильном приложении могут быть реализованы следующие пространственные инструменты:

— вычисление Евклидовых расстояний, что позволяет строить маршруты и показывать следование по ним в режиме реального времени благодаря встроенным в мобильные устройства GPS-приемникам (инструмент навигации), а также отображать длину построенного маршрута;

— запись построенных треков;

— генерализация – отображение на карте интересующих объектов, очищая ее от таких объектов, которые в рамках имеющейся задачи не представляют интереса;

— построение буферных зон и зон поиска;

— добавление новых векторных данных на карту, а также их редактирование или удаление.

6.3 Реализация мобильной ГИС

В данной работе реализовано нативное мобильное приложение, демонстрирующее работу инструментов пространственного анализа. Разработанные инструменты универсальны, и могут быть использованы в мобильных приложениях любой тематики.

Нативное приложение разработано на языке программирования JavaScript с использованием фреймворка React Native для разработки кроссплатформенных приложений для Android и iOS. Таким образом, логические компоненты приложения одинаковые, но интерактивные компоненты (такие как навигация, меню) обычно разные и должны разрабатываться для каждой операционной системы отдельно.

Для отображения карты и разработки инструментов пространственного анализа использована библиотека React-Native-Map. Она предоставляет компоненты для построения инструментов пространственного анализа в мобильном приложении.

Перед началом разработки необходимо установить на компьютер, на котором будет вестись установка, программную платформу NodeJS, интерпретирующую язык JavaScript из узкоспециализированного языка в язык общего назначения и инициализировать новый проект командой npm init в консоли, находясь в корневой папке проекта. Npm – это менеджер пакетов, входящий в состав Node.js. Затем, с помощью менеджера npm необходимо произвести установку пакетов Webpack, React, React Native, и React-Native-Maps запустив в консоли команды npm i X, где X – имя пакета. Webpack – файловый менеджер сборки проекта. В корневой папке проекта будут добавлены необходимые для работы приложения конфигурации и файл входной точки App.js. Вести разработку необходимо в этом файле, либо подключать другие файлы в нем.

Отладка приложения ведется на устройстве с операционной системой Android. Чтобы запустить ее, необходимо установить утилиту Adb,

предварительно установив на компьютер с официального сайта-поставщика. Мобильное устройство необходимо подсоединить к компьютеру, на котором ведется разработка и в настройках устройства разрешить отладку по USB. Запустить проект командой react-native run-android.

Приложение содержит 644 строки кода, не включая конфигурационные файлы сборщика приложений Webpack. Объем, который приложение занимает на дисковом пространстве – 78 Мб в режиме разработки, 4 Мб – в режиме сборки. Приложение может быть запущено под операционной системой Android. Разработка проводилась на персональном четырехъядерном ноутбуке MSI, CPU – Intel Core i7-7700HQ, объем ОЗУ – 8 Гб под управление 64-разрядной операционной системой Windows Professional.

Тематика разработанного мобильного приложения – медицина. Пользователю доступна карта города Красноярска с медицинскими учреждениями, также на карте указано непосредственно местоположение самого пользователя. Доступные функции в приложении:

— отображение местоположения пользователя;

— получение координат местоположения пользователя;

— отображение медицинских учреждений;

— отображение информационного окна для выбранного медицинского учреждения;

— построение зоны поиска для выборки медицинских учреждений, попавших в зону поиска;

— построение пешеходного, автомобильного и автобусного маршрутов к

— выбранному медицинскому учреждению;

— запись на прием в выбранное медицинское учреждение;

— фильтрация отображения медицинских учреждений по типу;

— добавление выбранного медицинского учреждения в список Избранных медицинских учреждений;

— просмотр списка Избранных медицинских учреждений и приближение на карте к выбранному учреждению;

Данные предоставлены порталом Енисей-ГИС в формате CSV и распространяются на свободной основе.

В разделах 6.3.1 – 6.3.3 описана реализация вышеописанного функционала, включающая в себя его исходный код и иллюстрации работы в мобильном приложении.

6.3.1 Инструменты работы с местоположением пользователя

При запуске приложения пользователю показывается главный экран, на котором показаны медицинские учреждения и местоположение пользователя. Красными маркерами обозначены медицинские учреждения, синим – местоположение пользователя. На рисунке 8 представлен начальный экран мобильного приложения.

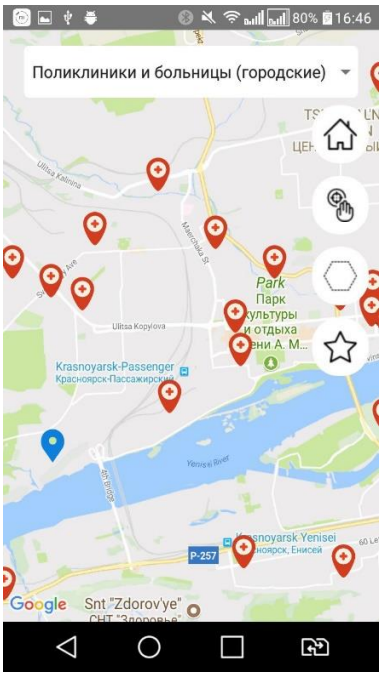

Рисунок 8 – Начальный экран мобильного приложения

В верхней части экрана расположено поле выбора типа медицинских учреждений, который отображается на карте. В правой части экрана расположены кнопки с инструментами. Сверху вниз – кнопка «Домой»,

«Получить мое местоположение», «Построить зону поиска», «Список избранных учреждений». Код компонентов карты, иконок и кнопок показан в листинге 1 приложения В.

При нажатии на кнопку «Получить мое местоположение» в буфер обмена копируется специальная ссылка с координатами пользователя, которую можно открыть в браузере. Такой способ предпочтительнее и нагляднее простой отправки координат.

6.3.2 Инструмент построения зоны поиска

Инструмент построения зоны поиска позволяет пользователю вручную определить область интереса на карте, внутри которой его интересуют медицинские учреждения. После нажатия на кнопку «Построить зону поиска» пользователь нажатиями пальцев по карте ставит точки, которые соединяются между собой, образуя зону поиска.

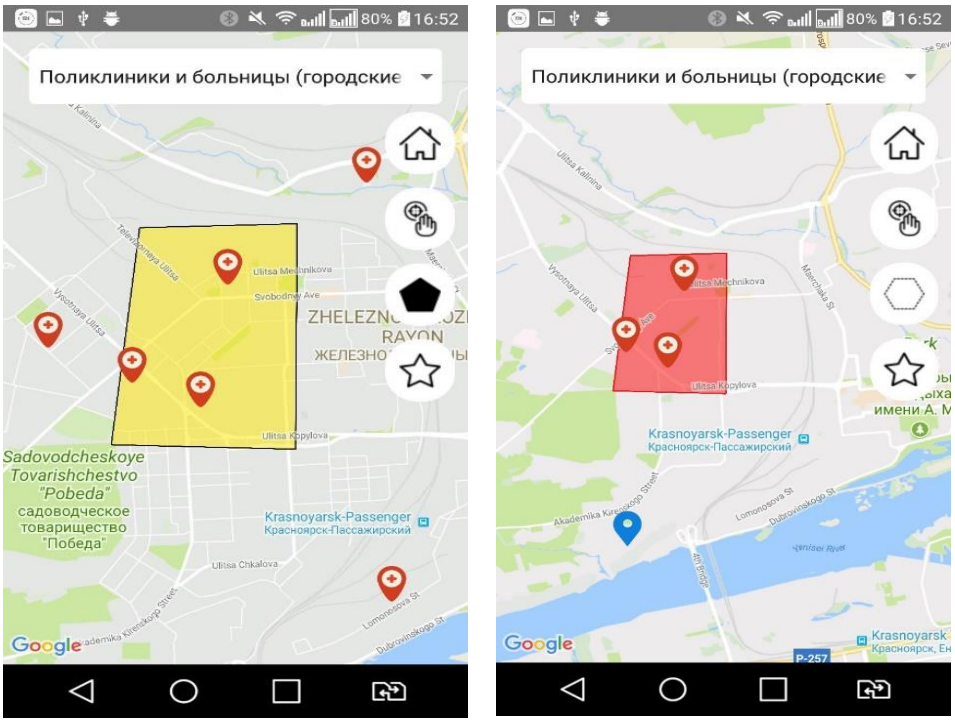

Рисунок 9 – а) построение зоны поиска, б) построенная зона поиска

Листинг программного кода построения зоны поиска приведен в приложении В.

Инструмент полезен, когда на карте одновременно отображается слишком много объектов разных категорий, а переключение видимости отдельных слоев с категориями может скрыть часть интересующих объектов.

6.3.3 Инструмент построения маршрутов и информационные блоки

При нажатии на иконку интересующего пользователя медицинского учреждения в нижней части экрана появляется информационное окно, при этом иконка становится большего размера. Данное информационное окно содержит в себе название медицинского учреждения, иконку, показывающую имеется ли учреждение в списке «Избранные медицинские учреждения», иконку записи на прием, а также три иконки построения маршрута к данному учреждению (пешего, автобусного или автомобильного).

При добавлении учреждения в список «Избранные медицинские учреждения» иконка окрашивается в желтый цвет и появляется в соответствующем списке, который доступен к просмотру при нажатии на кнопку «Показать избранные медицинские учреждения».

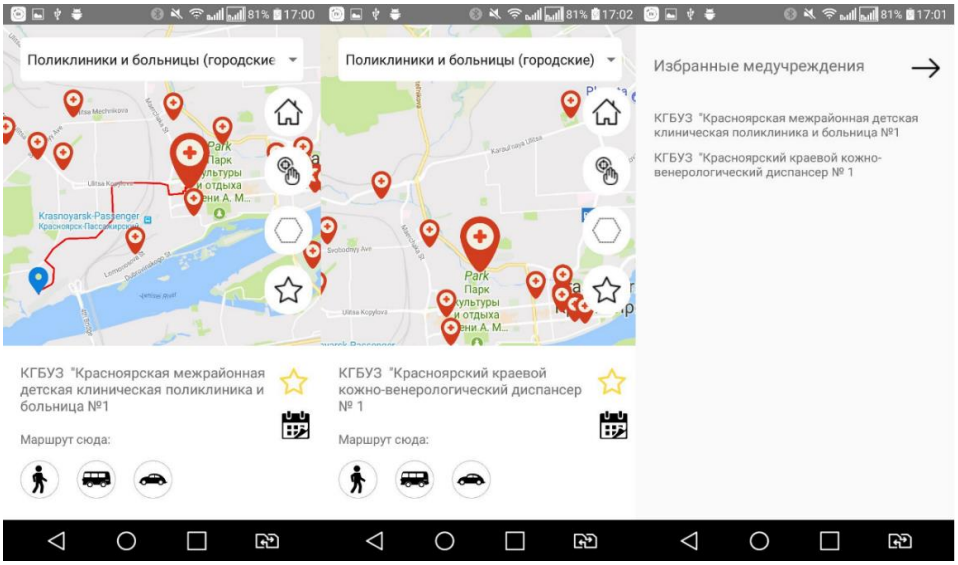

Рисунок 10 – Список избранных медицинских учреждений

Чтобы занести учреждение в список избранных медицинских учреждений, необходимо нажать на кнопку с иконкой звезды.

При записи на прием пользователю необходимо нажать на иконку календаря. При этом сработает переход в браузер, выбранный по умолчанию на устройстве пользователя, где откроется страница с онлайн-записью на прием через сервис «Веб-Регистратура.ру». Каждое учреждение имеет уникальный номер, который используется для формирования ссылки в браузере.

Для построения маршрутов к медицинскому учреждению необходимо нажать на интересующий вид маршрута: пешеходный, автобусный, автомобильный. В результате выбора на карте строится линия красного цвета, которая является маршрутом от местоположения пользователя до выбранного медицинского учреждения.

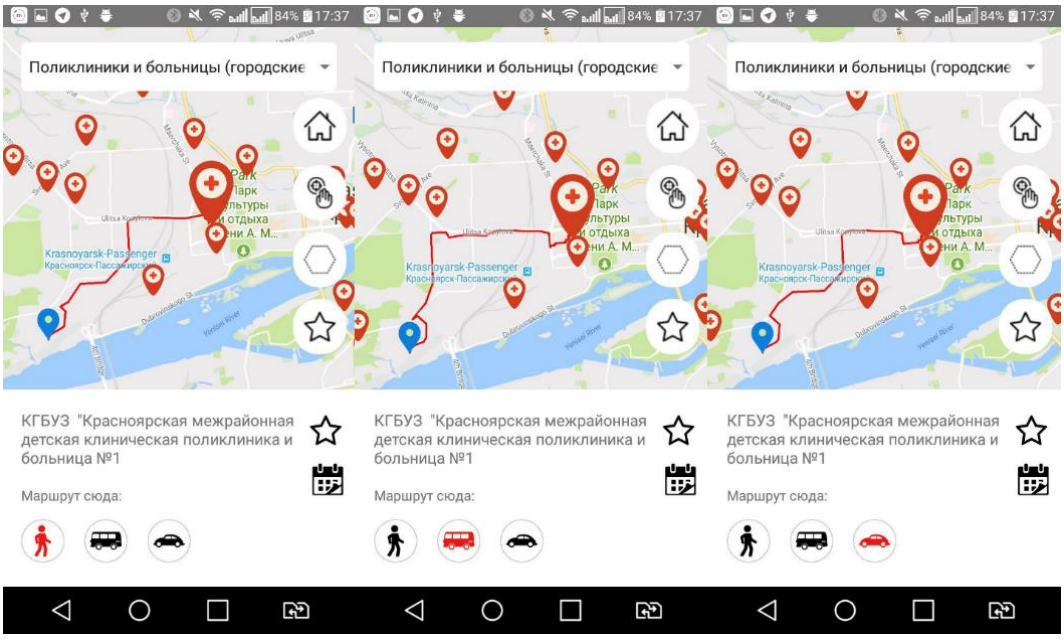

Рисунок 11 – Построение маршрутов до выбранного медицинского учреждения

Фильтрация медицинских учреждений по типу производится при нажатии на поле выбора типов в верхней части экрана. После выбора интересующего типа на карте останутся только те учреждения, которые относятся к этому типу.

7 Разработка веб-ГИС

В данной главе описаны методики и технологии проектирования и конструирования веб-ГИС. Положительной стороной разработки веб-ГИС является возможность реализовать более широкий функционал, по сравнению с картографическими мобильными приложениями. Такой функционал может включать в себя сложные пространственные операции, оказывающие большую нагрузку на процессор компьютера, но при этом позволяющие выполнять более глубокие и полезные манипуляции с данными.

7.1 Исходный пакет инструментов для разработки веб-ГИС

В качестве инструментов для разработки веб-ГИС могут использоваться любые ГИС-серверы (см. раздел 3), ГИС-сервисы (см. раздел 4) и системы управления базами данных (см. раздел 5). Особенно удобны для разработки веб-ГИС клиентские приложения (см. раздел 4.1.5), так как разработка ведется непосредственно либо в браузере, либо с использованием настольного клиентского приложения. Зачастую такой вид инструментов изначально создан для разработки именно веб-ГИС. Преимуществами веб-ГИС является независимость от операционных систем и браузеров (при этом при разработке нужно сделать упор на кроссбраузерность приложения), общий доступ к децентрализованным источникам данных, отсутствие необходимости установки дополнительного программного обеспечения для работы с веб-ГИС. Среди недостатков можно отметить зависимость от подключения к сети Интернет, необходимость иметь достаточно мощный компьютер для стабильного использования всего функционала веб-ГИС, а также затраты на разработку веб-ГИС.

Следует отметить, что веб-ГИС зачастую является классическим «Rich Internet Application (RIA)», т.е. приложением, доступным через Интернет и обладающим функциональностью традиционных настольных приложений. Как

правило, приложение RIA передает браузеру необходимую часть пользовательского интерфейса, оставляя большую часть данных (ресурсы программы, данные и т.д.) на сервере, что позволяет более сбалансированно использоваться вычислительные ресурсы клиента и сервера и обрабатывать большее количество сессий одновременно за счет одного и того же программного обеспечения [45]. Также существуют веб-ГИС в виде «Single Page Application (SPA)», когда практически все вычисления производятся на клиенте, а серверная часть служит лишь для обработки и отдачи данных клиенту. Такие приложения решают проблему необходимости иметь мощный компьютер для использования веб-ГИС, но при этом функционал таких приложений не должен быть слишком сложным в плане вычислений.

7.2 Языковые средства программирования для разработки веб-ГИС

Все приложения делятся на две части – клиентскую и серверную. Клиентская часть веб приложения – это графический интерфейс. Именно графический интерфейс отображается в браузере. Пользователь взаимодействует с веб-приложением через браузер, кликая по ссылкам и кнопкам. Серверная часть веб-приложения – это программа или скрипт на сервере, обрабатывающая запросы пользователя (которые инициализируются браузером). Сервер обрабатывает запросы, вызывая специальный скрипт для обработки, который формирует веб-страницу, и отсылает клиенту по сети. Браузер тут же отображает полученный результат в виде очередной веб-страницы.

Основной язык, которым описывается графический интерфейс веб-приложения – это HTML. Этот язык описывает структуру веб-страницы. Художественное оформление веб страниц описывается таблицами стилей – CSS.

Для добавления функциональности графическому интерфейсу используются дополнительные технологии: скрипты, написанные на языке программирования JavaScript. В настоящий момент существует множество

фреймворков для разработки сложных веб-ГИС. Такие фреймворки позволяют шаблонизировать структуру приложения, позволяя разработчику реализовывать непосредственно функционал веб-ГИС, оставляя низкоуровневые операции по управлению памятью и т.п. фреймворку. Самыми распространенными фреймворками являются ReactJS, Angular, Vue. Все они написаны на языке JavaScript. Зачастую такие приложения состоят из компонентов, тем самым реализуя пользовательский интерфейс с богатыми возможностями.

Для программирования серверной части веб приложения может использоваться большое количество языков программирования, такие как PHP, Perl, Python, Java, Ruby, Go, NodeJS. Независимо от языка, на котором написана серверная часть веб-приложения, способы обработки запросов и взаимодействия с пользователем остаются одинаковыми. Зачастую предпочтителен к применению тот же язык программирования, на котором написан ГИС-сервер, если разработчику необходимо самостоятельно конфигурировать ГИС-сервер [46].

7.3 Особенности UI/UX-проектирования и конструирования клиентской части веб-ГИС

При разработке интерфейса клиентской части веб-ГИС целесообразно пользоваться дизайн-системами известных компаний и спецификациями, например, спецификацией от Google Material Design (см раздел 6.1.2).

В настоящее время много различных веб-ГИС, предоставляющих различную информацию. Они, прежде всего, ориентированы на картографическую информацию, но это их никак не выделяет на общем фоне подобных ресурсов и веб-приложений. Чтобы пользователь запомнил веб-приложение и выделил его из большого количества других, оно должно быть в чем-то уникальным. Это может быть удобство, специфичные функции, средства пространственного анализа и т.д. Допустим, веб-ГИС ориентирована на туризм и зоны отдыха. Пользователь заходит на ресурс и видит множество мест,

предлагаемых системой, их адреса и как к ним проехать, но как он может сделать выбор, если все их он видит впервые?

Первый вариант: он может опросить несколько десятков случайных прохожих, что они думают о том или ином месте отдыха в течение нескольких дней. Но, их мнения не вызовут у него никакого доверия, так как эти люди ему абсолютно незнакомы.

Второй вариант: пользователь может сэкономить массу времени, прочитав отзывы и оценки о различных местах отдыха на ресурсе веб-ГИС, выбирая указанное место на карте. Стоит отметить, что в Интернете доверие к незнакомым людям в разы выше, чем в реальности. Пользователи будут читать отзывы и определят, куда и когда они поедут отдыхать. Почему? Ответ прост: в реальности на мнение, которое выскажет человек, влияет множество факторов, начиная от окружающей обстановки и заканчивая оппонентом. В то время как в Интернете они просто оставляют отзыв, не считаясь с тем, кто и при каких обстоятельствах это будет читать.

Отзывы и рейтинги влияют на подсознание пользователя в рамках механизма социального одобрения. И также придают рациональную окраску его интуитивному выбору. Какие-либо технические аспекты гостиничных комплексов или отелей, графики уровня обслуживания персонала и т.п. оставляют шанс пользователю сказать самому себе, что он принял мудрое осмысленное решение. Наиболее эффективны рейтинги и оценки, в которых есть какой-то рассказ. По мнению психологов, это создает впечатление личного знакомства с рассказчиком и повышает доверие к его словам [47].

При конструировании клиентской части с использованием языка HTML целесообразно воспользоваться методологией БЭМ. БЭМ (Блок, Элемент, Модификатор) — компонентный подход к веб-разработке, разработанный компанией Яндекс. В его основе лежит принцип разделения интерфейса на независимые блоки, которые впоследствии могут быть использованы в любом другом веб-приложении. БЭМ включает в себя:

— методологические рекомендации по разработке сайтов — простые советы по организации проекта, который нужно сделать быстро, а поддерживать долгие годы;

— технологии и библиотеки с открытым исходным кодом — готовая реализация рекомендаций БЭМ;

— инструменты для автоматизации работы с методологией БЭМ.

Возможности БЭМ:

— простая поддержка структуры кода при росте проекта;

— повторное использование кода;

— точечные изменения с минимальными затратами: обновление дизайна, добавление функциональных элементов и т. д. [48]

7.4 Инструменты пространственного анализа в веб-ГИС

Построение буфера создает область определенного радиуса вокруг выбранных объектов – зону поиска. Для построения зоны поиска в веб-ГИС может быть использовано сочетание инструментов построения геометрических объектов (окружность, прямоугольник, многоугольник) и вычисления принадлежности объектов ограниченной зоне. Последний инструмент может быть реализован путем вычислений на основе координат зоны поиска и объектов, попадающих в нее. Объект также может иметь более одной зоны поиска.

Построение буферов и зон поиска – это важный и часто используемый инструмент пространственного анализа, но существуют многие другие инструменты, которые так же используются в веб-ГИС. Пространственное наложение – это процесс, позволяющий выявить пространственные взаимоотношения между двумя полигональными слоями, имеющими общие участки. Такими взаимоотношениями могут быть принадлежность, пересечение, объединение, разница объектов. Инструмент пространственного наложения также может быть реализован при помощи координатных вычислений.

Использование известных значений той или иной величины в определенных точках для оценки неизвестных значений в неизвестных точках называется пространственной интерполяцией. Пространственная интерполяция в веб-ГИС вычисляется путем подсчета выбранного количественного параметра интересующих объектов. Затем строятся интервалы, такие, что выбранные значения можно было бы объединить в одну группу (к примеру, значения 0 – 10 – низкая температура, 11 – 20 – средняя, 21 – 30 высокая). Далее с учетом произведенных вычислений над количественными параметрами и интервалами области вокруг рассматриваемых объектов окрашиваются в соответствующий цвет.

7.5 Реализация веб-ГИС

В данной работе реализована веб-ГИС с использованием языка программирования JavaScript и библиотеки jQuery, архитектура приложения – SPA. В качестве сервера использована связка – ГИС-сервер GeoServer и сервер Apache Tomcat 8.5, при создании веб-ГИС использовалась библиотека Openlayers 4. Слой медицинских учреждений хранится в СУБД PostgreSQL 9.5.

Веб-ГИС содержит 1203 строки кода. Объем, который приложение занимает на дисковом пространстве – 166 Кб. Веб-ГИС может быть запущена в любом браузере. Разработка проводилась на персональном четырехъядерном ноутбуке MSI, CPU – Intel Core i7-7700HQ, объем ОЗУ – 8 Гб под управлением 64-разрядной операционной системой Windows Professional.

Тематика разработанной веб-ГИС – медицина. Пользователю доступна карта города Красноярска с медицинскими учреждениями, также на карте указано непосредственно местоположение самого пользователя. Функционал приложения идентичен разработанному в данной работе мобильному приложению с ГИС-инструментарием (см. раздел 6.3), также добавлены инструменты измерения расстояний, построения тематических карт, печати тематических карт и добавления собственных заметок к избранным

медицинским учреждениям. Данные предоставлены порталом Енисей-ГИС в формате CSV и распространяются на свободной основе.

В разделах 7.5.1 – 7.5.5 описана реализация функционала веб-ГИС, включающая в себя его исходный код и иллюстрации работы веб-ГИС в браузере.

7.5.1 Инструменты работы с местоположением пользователя в веб-ГИС

При запуске веб-ГИС пользователю показывается главный экран, на котором расположены кнопки рисования зоны поиска, измерения расстояний, получения местоположения пользователя, просмотра списка избранных медицинских учреждений и печати карты, а также инструмент масштабирования, легенда и шкала масштаба. Главный экран представлен на рисунке 12.

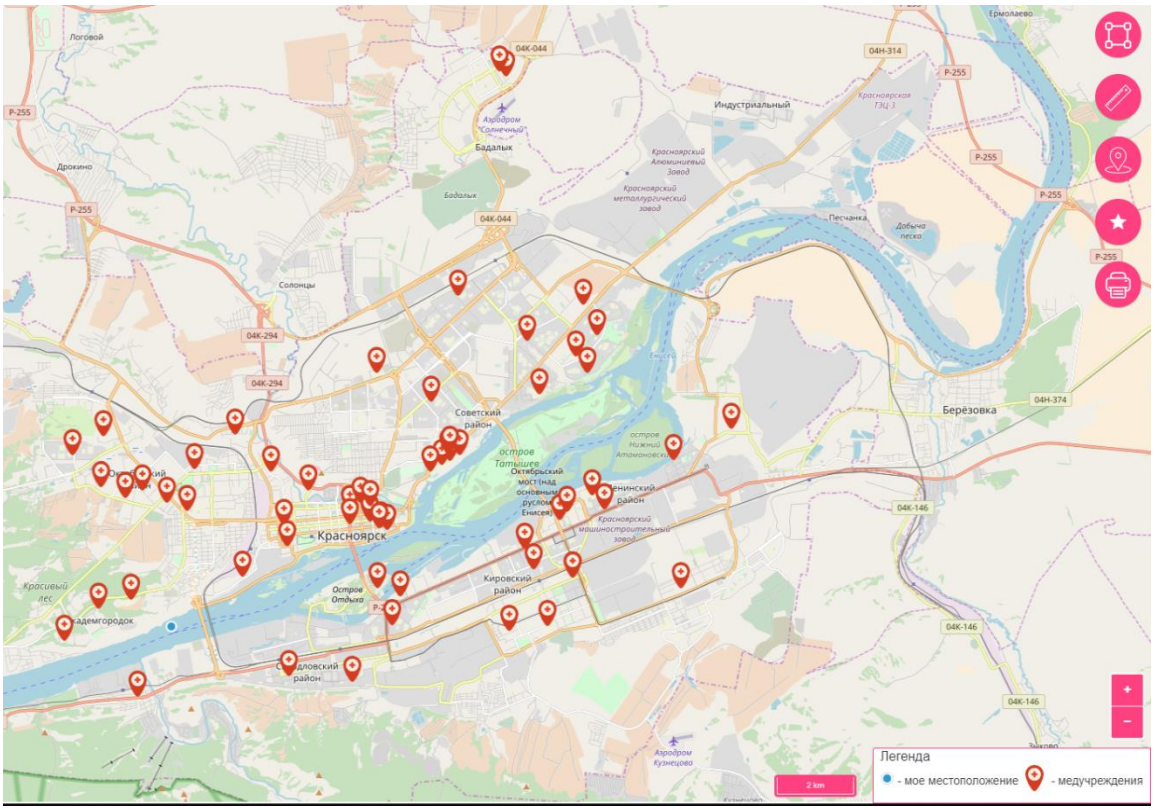

Рисунок 12 – Главный экран веб-ГИС

При клике на кнопку получения местоположения пользователя появляется информационное окно, сообщающее пользователю о том, что его координаты успешно скопированы в буфер обмена. Синей точкой на карте указано местоположение пользователя, при этом, прежде чем его показать, браузер запросит у пользователя доступ к местоположению. Копирование координат происходит не в чистом в виде, а в виде ссылки на ресурс Google Карты для удобства использования. Листинг программного кода получения местоположения пользователя указан в приложении Г (см. раздел 1).

7.5.2 Зона поиска и обзор информационной панели

При нажатии на кнопку рисования зоны поиска пользователю предлагает выбор формы рисования во всплывающем окне. Имеется несколько вариантов на выбор – окружность, прямоугольник, многоугольник, квадрат. При выборе одной из форм пользователю необходимо нарисовать на карте зону поиска. Процесс рисования зоны поиска показан на рисунке 13.

Рисунок 13 – Процесс рисования зоны поиска формы окружности

После окончания рисования зоны поиска в левой части веб-ГИС пользователю показывается информационная панель со всеми медицинскими учреждениями, попавшими в зону поиска. При этом у объектов, попавших в зону поиска, размер иконки увеличивается в 2 раза. На рисунке 14 представлено информационная панель со всеми медицинскими учреждениями, попавшими в зону поиска.

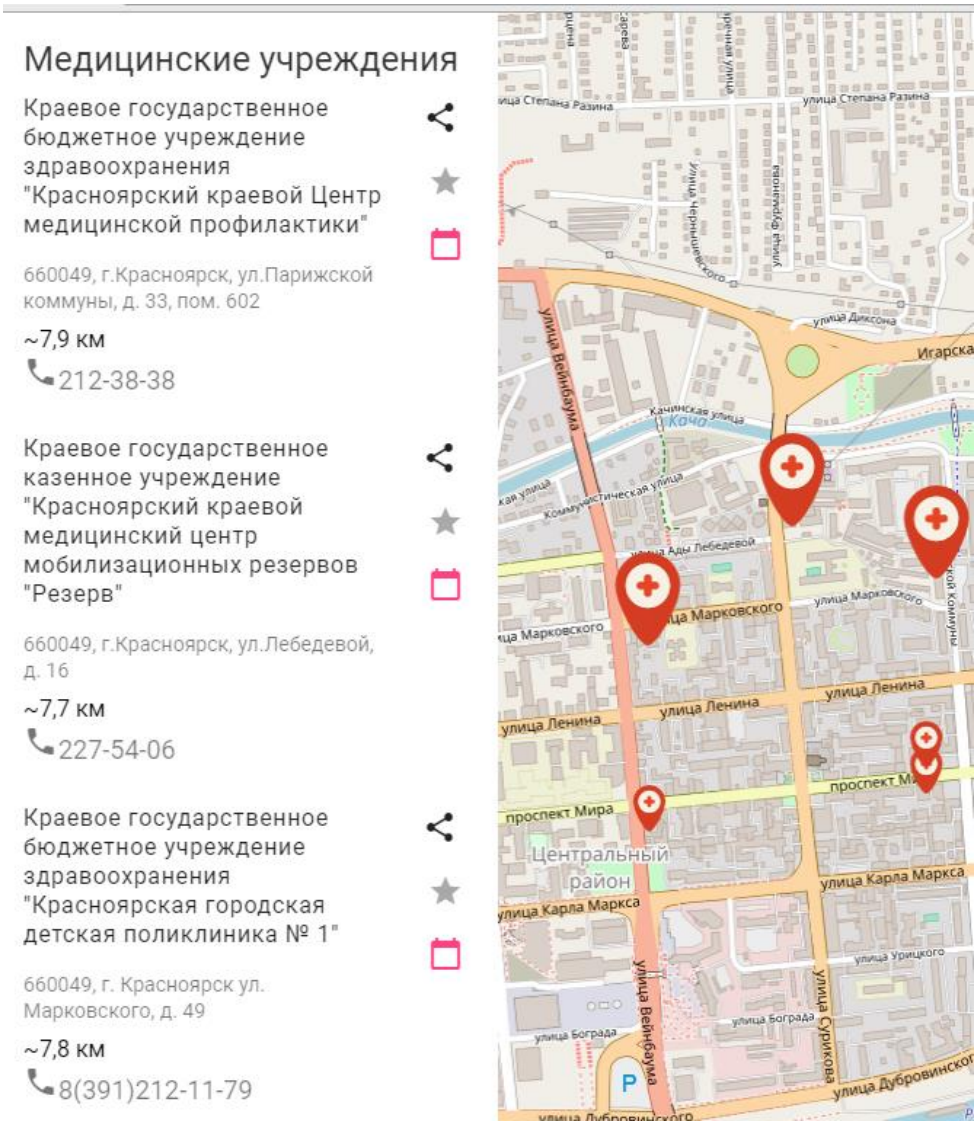

Рисунок 14 – Информационная панель со всеми медицинскими учреждениями, попавшими в зону поиска

Как видно из рисунка 14, информационная панель включает в себя название объекта, его адрес, примерное расстояние от местоположения пользователя и телефон. Также в правой части информационного окна показаны действия, которые можно совершить с медицинским учреждением: рассказать о нем в социальных сетях, добавить в список Избранных медучреждений, записаться на прием в выбранное медучреждение. Листинг программного кода построения зоны поиска и создания информационной панели указан в приложении Г (см. раздел 2).

7.5.3 Инструменты работы с медицинскими учреждениями

При появлении информационной панели у пользователя появляется возможность добавить медицинское учреждение в список Избранных медицинских учреждений посредством нажатия на иконку добавления в список Избранных медицинских учреждений, при этом цвет иконки изменится в розовый цвет. Просмотреть список можно при нажатии на кнопку «Избранные медицинские учреждения» на главном экране веб-ГИС (см. рисунок 15).

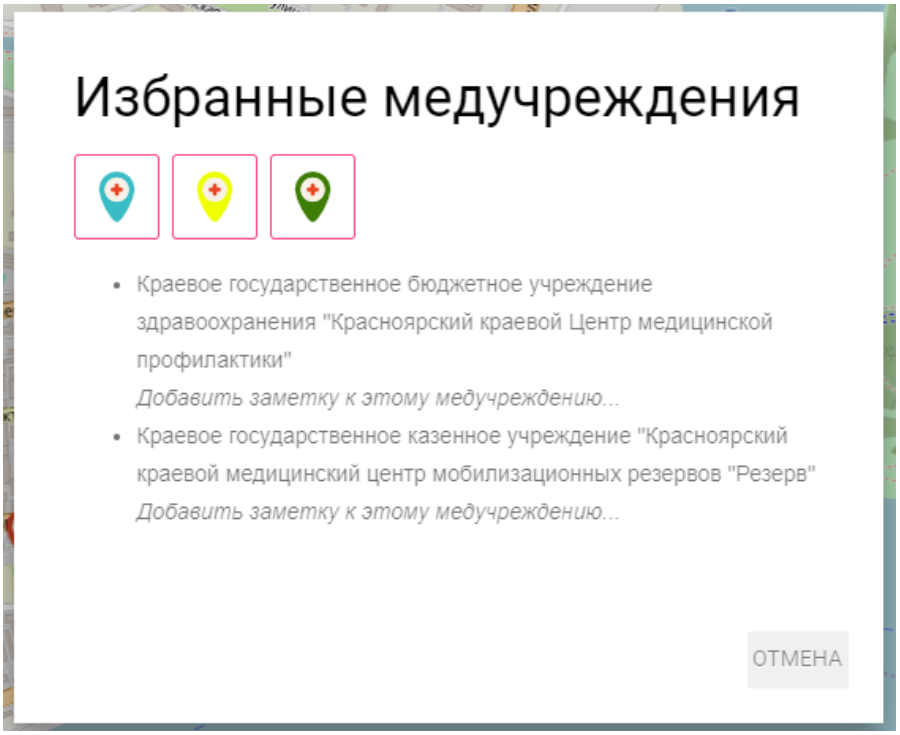

Рисунок 15 – Модальное окно «Избранные медучреждения»

Пользователю показывается модальное окно со списком избранных медицинских учреждений. Также под заголовком расположены три кнопки с иконками разных цветов. Эти кнопки позволяют изменить цвет иконки для избранных медицинских учреждений, таким образом пользователь может построить тематическую карту. Листинг построения тематической карты указан в приложении Г (см. раздел 3). Также пользователю предоставляется возможность добавить собственную заметку к избранному медицинскому

учреждению, для этого необходимо нажать на ссылку «Добавить заметку к этому медучреждению…». В появившемся окне пользователь может сделать запись, и она будет отображаться в информационной панели при выборе данного медицинского учреждения. Листинг добавления заметки представлен в приложении Г (см. раздел 4).

Также пользователь может рассказать о медицинском учреждении в социальных сетях, для этого необходимо нажать на иконку «Рассказать о медучреждении». При нажатии на нее открывается модальное окно, в котором пользователю необходимо выбрать, в какой социальной сети он хочет поделиться информацией о медицинском учреждении. Информация содержит в себе ссылку на медицинское учреждение в ресурсе Google Карты, а также текстовое дополнение «Рекомендую обратить внимание на эту клинику!». Листинг данного инструмента указан в приложении Г (см. раздел 5). Модальное окно «Рассказать о медучреждении» показано на рисунке 16.

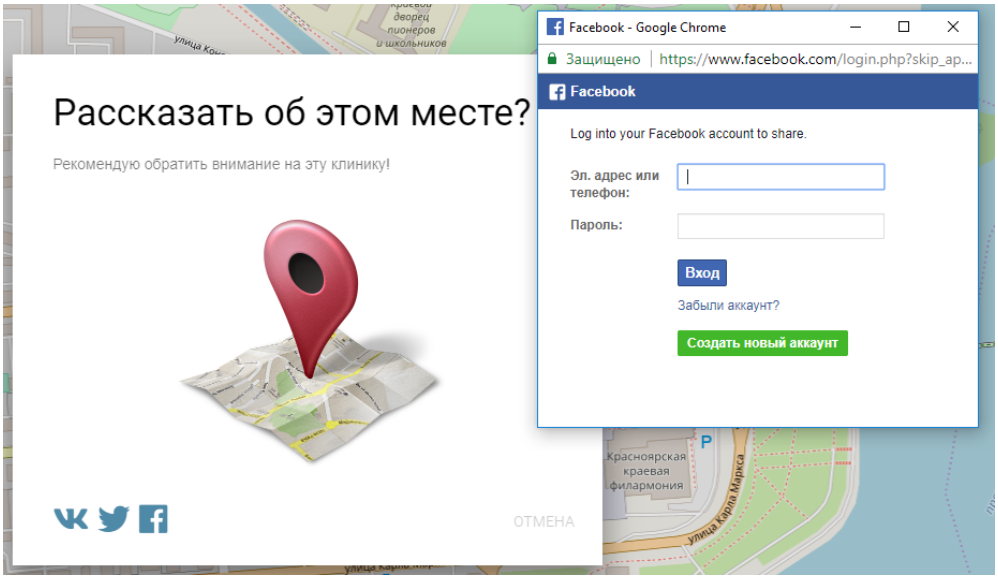

Рисунок 16 – Модальное окно «Рассказать о медучреждении» и окно входа в аккаунт Facebook

При клике на иконку записи на прием пользователь будет перенаправлен на новую вкладку в браузере, где открывается сайт «Веб-Регистратура.ру».

Благодаря уникальному идентификатору каждого медицинского учреждения окно записи будет открыто именно для того медицинского учреждения, напротив которого был произведено нажатие по иконке.

7.5.4 Инструмент измерения расстояний

Пользователю доступен инструмент измерения расстояний при нажатии на кнопку «Измерение расстояний» на главном экране веб-ГИС. Расстояние измеряется в метрах и в километрах. При нажатии на данную кнопку пользователь может сразу же начинать строить на карте линию, с продолжением которой будет появляться и обновляться информационное всплывающее окно с измеренным расстоянием. Процесс измерения расстояния в веб-ГИС показан на рисунке 17.

Рисунок 17 – Измерение расстояния между двумя медучреждениями

Листинг инструмента «Измерение расстояний» указан в приложении Г (см. раздел 6).

7.5.5 Инструмент печати тематических карт

Как было сказано в разделе 7.5.3 данной работы, пользователь может строить тематические карты, выделяя на них медицинские учреждения посредством занесения их в список Избранных медицинских учреждений. После

выбора избранных медицинских учреждений легенда в нижнем правом углу веб-ГИС обновляется в соответствии с выбранной иконкой в модальном окне «Избранные медицинские учреждения». Далее для подготовки тематической карты к печати необходимо нажать на кнопку «Печать», расположенную на главном экране веб-ГИС. Подготовленная к печати карта показана на рисунке 18.

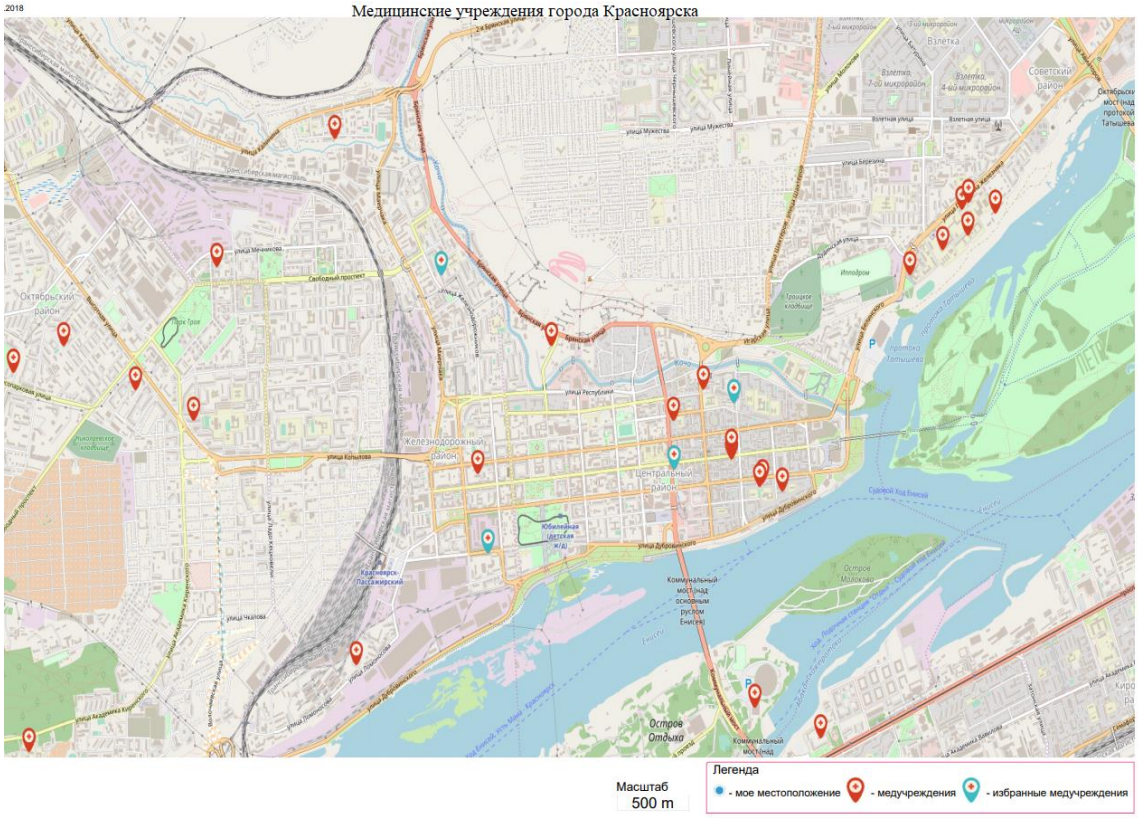

Рисунок 18 – Тематическая карта медицинских учреждений города Красноярска, подготовленная к печати

Листинг изменения иконки избранных медицинских учреждений и подготовки тематической карты к печати указан в приложении Г (см. раздел 3).

ЗАКЛЮЧЕНИЕ

В ходе работы рассмотрена область веб-картографии и создания мобильных и веб-ГИС, а также приведены примеры мобильных и веб-ГИС. Составлен список актуальных определений и терминов в области веб-картографии, некоторые из них описаны в различных ГОСТ, а некоторые приведены из других источников (статьи, учебники), либо сформулированы самостоятельно автором данной работы. Рассмотрен инструментарий для создания мобильных и веб-ГИС, а именно ГИС-серверы, ГИС-сервисы, системы управления базами данных, а также приведены результаты их тестирования. Все результаты тестирований сгруппированы и объединены для последующего сравнительного анализа при выборе конкретного инструмента для разработки. По итогам всех тестирований можно сделать вывод, что выбор инструмента будет зависеть от целей разработки, типа обрабатываемых данных, возможностей, которые необходимый инструмент предоставляет.

В результате работы спроектированы и сконструированы мобильная и веб-ГИС, обладающие минимальным необходимым ГИС-инструментарием. Составлено методическое обеспечение для разработки мобильных и веб-ГИС, включающее в себя методики проектирования интерфейса и построения архитектуры приложений. Данными методиками можно воспользоваться при проектировании архитектуры мобильных и веб-ГИС, а приведенные в работе листинги с программным кодом можно использовать при создании собственного ГИС-инструментария в разрабатываемом приложении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Аникеева О.С. Публикация карт в сети интернет: эволюция картографии / О.С. Аникеева // Журнал Северо-Кавказского Федерального университета Наука. Инновации. Технологии. – 2015. – № 2. – С. 78 – 85.;

2 Mozilla MDN Web Docs [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Pages_sites_servers_and_search_engines

3 Хабр. Статья «Веб-сервисы в теории и на практике для начинающих» [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/post/46374/;

4 Берлянт А.М. Картография. Учебник для ВУЗов / М.: Аспект Пресс. – 2002. – 336 с.;

5 ArcGIS Desktop. Геокодирование [Электронный ресурс]. – Режим доступа: https://pro.arcgis.com/ru/pro-app/help/data/geocoding/what-is-geocoding-.htm;

6 Совзонд [Электронный ресурс]. – Режим доступа: https://sovzond.ru/services/gis/services/;

7 Капралов Е.Г., Кошкарев А.В., Тикунов В.С. и др. Геоинформатика в 2 кн. 2 кн. / Учебник для студ. высш. уч. заведений. – 2010. – 432 с.;

8 Трипутина В.В. Моделирование и разработка ГИС-сервисов для задач исследований в области энергетики / Журнал Вычислительные технологии. – 2008. – Т.1 - №S1. – С. 78-87;

9 «Хабр». «Статья Что такое UX/UI дизайн на самом деле?» [Электронный ресурс]. – Режим доступа: https://habr.com/post/321312/;

10 Матин С.В., Круглов А.Н. Разработка клиентского модуля для анализа и визуализации пространственных данных в картографических веб-подсистемах сервис-ориентированных корпоративных систем управления // Журнал Энергетика. Инновационные направления в энергетике. Cals-технологии в энергетике. – 2014. – № 1 – С. 316 – 324.;

11 Российская сеть изучения и охраны пернатых хищников. Статья «Веб-ГИС «ФАУНИСТИКА», как пример успешного краудсорсинг-проекта в деле

изучения и охраны птиц» [Электронный ресурс]. – Режим доступа: http://rrrcn.ru/ru/archives/24578;

12 NextGIS. Статья «ГИС для оперативного контроля строительства волоконно-оптических линий связи (ВОЛС)» [Электронный ресурс]. – Режим доступа: http://nextgis.ru/blog/compulink/;

13 Гурин Е.А., Бронзова Ж.Е. Increase of the computer perfomance by parallelization // Журнал Академическая Публицистика. – Изд.: Общество с ограниченной ответственностью "Аэтерна" (Уфа) – 2018. – №2. – С. 6 – 12.;

14 Открытые системы. Статья «Закон Амдала и будущее многоядерных процессоров» [Электронный ресурс]. – Режим доступа: https://www.osp.ru/os/2009/04/9288815/;

15 IXBT. Статья «Влияние параметров памяти на производительность системы» [Электронный ресурс]. – Режим доступа: https://www.ixbt.com/mainboard/memory-measuring-2011.shtml;

16 Ecological Informatics. Статья «Vector and raster - advantages and disadvantages» [Электронный ресурс]. – Режим доступа: http://planet.botany.uwc.ac.za/nisl/gis/gis_primer/page_19.htm;

17 GIS Geography. Статья «Vector vs Raster: What’s the Difference Between GIS Spatial Data Types?» [Электронный ресурс]. – Режим доступа: https://gisgeography.com/spatial-data-types-vector-raster/;

18 Конспект лекций по предмету «ГИС в геодезии». Составитель С.Г. Шнитко [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/3972941/page:9/;

19 Документация QGIS 2.18. Статья «Растровые данные» [Электронный ресурс]. – Режим доступа: https://docs.qgis.org/2.18/ru/docs/gentle_gis_introduction/raster_data.html;

20 GeoManual. Статья «Искажения в картографических проекциях; их распределение; определение размеров искажений на картах» [Электронный ресурс]. – Режим доступа: http://geoman.ru/books/item/f00/s00/z0000060/st012.shtml;

21 Картолог. Статья «Выбор проекций» [Электронный ресурс]. – Режим доступа: http://kartolog.ru/spravochnye-materialy/vybor-proekcij/;

22 NextGIS. Статья «Работа с проекциями» [Электронный ресурс]. – Режим доступа: http://nextgis.github.io/webgis_course/6/python_proj.html;

23 OSGeo Live. Статья «Стандарты «Open Geospatial Consortium»» [Электронный ресурс]. – Режим доступа: https://live.osgeo.org/ru/standards/standards.html;

24 ArcGIS Resources. Статья «How the GIS server works» [Электронный ресурс]. – Режим доступа: http://webhelp.esri.com/arcgisserver/9.2/dotnet/manager/administration/how_gis_svr_works.htm;

25 GIS-Lab («ГИС Лаборатория»). Статья «Создание картографических сервисов с использованием MapServer» [Электронный ресурс]. – Режим доступа: http://gis-lab.info/qa/mapserver.html;

26 GIS-Lab («ГИС Лаборатория»). Статья «Начало работы с GeoServer» [Электронный ресурс]. – Режим доступа: http://gis-lab.info/qa/geoserver-begin.html;

27 MapSurfer.NET. Официальный сайт [Электронный ресурс]. – Режим доступа: http://mapsurfernet.com/;

28 ArcGIS Enterprise. Статья «Что такое ArcGIS Server?» [Электронный ресурс]. – Режим доступа: https://enterprise.arcgis.com/ru/server/latest/get-started/windows/what-is-arcgis-for-server-.htm;

29 OSGeo Live. Статья «QGIS Server» [Электронный ресурс]. – Режим доступа: https://live.osgeo.org/ru/overview/qgis_mapserver_overview.html

30 NEXTGIS. «Статья Знакомство с Mapnik» [Электронный ресурс]. – Режим доступа: http://gis-lab.info/qa/mapnik.html http://nextgis.github.io/webgis_course/3/mapnik_introduction.html#%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0_mapnik;

31 ThinkGEO. Официальный сайт [Электронный ресурс]. – Режим доступа: https://thinkgeo.com/gisserver;

32 MapsurferNET. Статья «Benchmarking Mapping Toolkits in Tile Seeding» [Электронный ресурс]. – Режим доступа: http://mapsurfernet.com/blog/benchmarking-mapping-toolkits-in-tile-seeding;

33 Geotux Tuxfamily. Статья «Web mapping client comparison v.6» [Электронный ресурс]. – Режим доступа: http://geotux.tuxfamily.org/index.php/en/component/k2/item/291-comparacion-clientes-web-v6;

34 Mapbender. Официальный сайт [Электронный ресурс]. – Режим доступа: http://www.mapbender.org/;

35 InteticsGEO. Статья «Commercial vs. Open Source: A comparison of GIS Software» [Электронный ресурс]. – Режим доступа: https://geo.intetics.com/geo-blog/commercial-vs-open-source-a-comparison-of-gis-software;

36 Документация MySQL [Электронный ресурс]. – Режим доступа: http://www.mysql.ru/docs/man/What-is.html;

37 Документация PostgresPRO [Электронный ресурс]. – Режим доступа: https://postgrespro.ru/media/2016/07/20/postgrespro_ibm.pdf;

38 Oracle официальный сайт. Статья «Oracle Spatial and Graph» [Электронный ресурс]. – Режим доступа: http://www.oracle.com/technetwork/database-options/spatialandgraph/overview/spatialandgraph-1707409.html;

39 IBM Knowledge Center. Статья «Getting started with spatial data» [Электронный ресурс]. – Режим доступа: https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.spatial.doc/ids_spat_008.htm;

40 Suprio Ray, Bogdan Simion, Angela Demke Brown. Jackpine: A Benchmark to Evaluate Spatial Database Performance – 2011. – 12 с.;

41 Geekbrains. Статья «Выбираем язык для разработки мобильных приложений» [Электронный ресурс]. – Режим доступа: https://geekbrains.ru/posts/mobile_apps_languages?utm_source=cityads&utm_mediu

m=cpa&utm_campaign=cityads&utm_content=courses&utm_term=30%2F09%2F2017&partner_id=cityads&click_id=7IbZ1N9LmGZEx9L&sub_id=5zcW

42 Лайфхакер. Статья «Что такое Google Material Design и как он изменит нашу жизнь» [Электронный ресурс]. – Режим доступа: https://lifehacker.ru/google-material-design/

43 GIS-Lab («ГИС Лаборатория»). Статья «Краткое введение в ГИС. Часть 10: Пространственный анализ растровых данных: интерполяция» [Электронный ресурс]. – Режим доступа: http://gis-lab.info/qa/gentle-intro-gis-10.html

44 Хабр. Статья «Дизайн-система. Определение понятия» [Электронный ресурс]. – Режим доступа: https://habr.com/company/mailru/blog/351726/

45 Гордов Е.П., Окладников И.Г., Титов А.Г. Использование веб-ГИС технологий для разработки информационно-вычислительных систем для анализа пространственно-привязанных данных // Вестник НГУ. – 2011. – Т. 9 выпуск № 4 – 9 С.

46 Лабака.ру. Статья «Структура веб-приложения» [Электронный ресурс]. – Режим доступа: http://labaka.ru/likbez/struktura-veb-prilozheniya

47 Мандругин В.В. Психологические аспекты интерфейса в проектировании дизайна веб-ГИС // Интерэкспо ГЕО-Сибирь-2012. - Новосибирск: СГГА, 2012. – Т.2. – С.61-64.

48 Яндекс.Технологии. Статья «БЭМ» [Электронный ресурс]. – Режим доступа: https://tech.yandex.ru/bem/

ПРИЛОЖЕНИЕ А

Тестирование ГИС-серверов

А.1 Тестирование загрузки ГИС-серверами тайловых слоев без перепроецирования

На рисунке А.1 представлен график результатов запусков тестов, оценивающих время, необходимое для рендеринга выбранного для тестирования shape-файла (400 МБ) без перепроецирования и тайловой подгрузки в небольшом масштабе (от 0 до 9 единиц).

Рисунок А.1 – Результат запуска рендеринга shape-файла с тайловой подгрузкой без перепроецирования, где ось абсцисс показывает время, которое потребовалось для рендеринга, а ось ординат – масштаб

В целом, ГИС-серверы Mapnik/TileMill и Mapserver показывают небольшое увеличение времени загрузки тайлов при масштабировании, в то время, как все остальные имеют достаточно большой скачок на графике при более высоком уровне масштабирования.

На рисунке А.2 представлен график результатов запусков тестов, оценивающих пропускную способность ГИС-серверов по отношению к количеству загруженных тайлов при загрузке shape-файлов без перепроецирования.

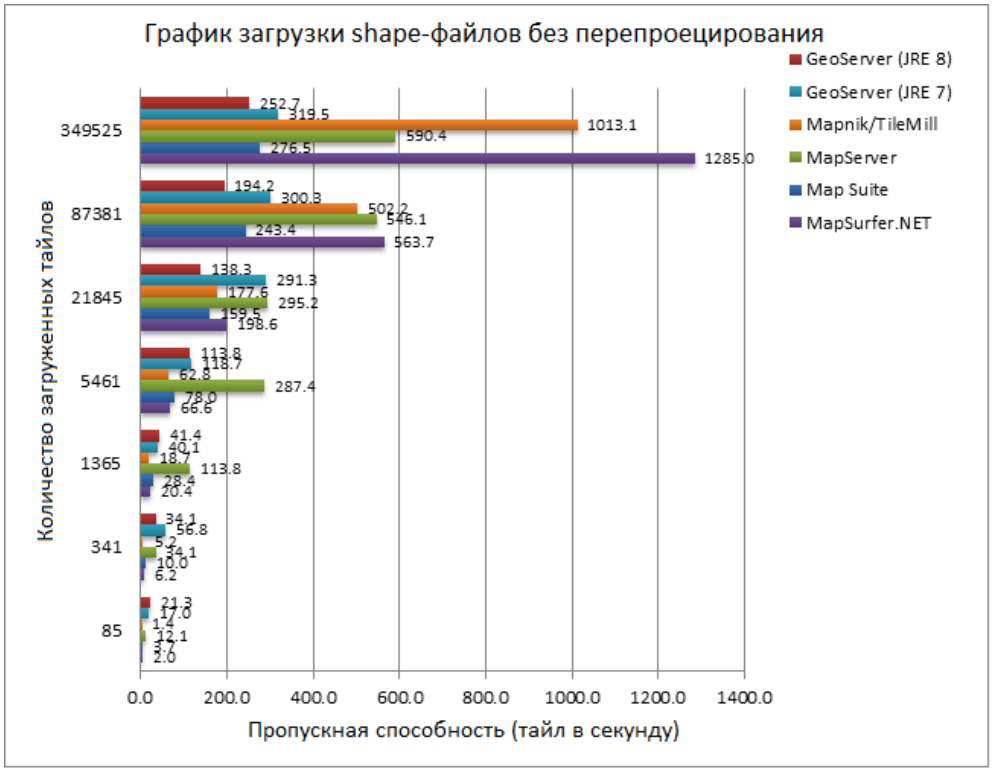

Рисунок А.2 – Результат запуска рендеринга shape-файла с тайловой подгрузкой без перепроецирования, где ось абсцисс показывает пропускную способность (тайл в секунду), а ось ординат – количество загруженных тайлов, которое потребовалось для рендеринга

Из рисунка А.2 видно, что ГИС-сервер MapSurfer.NET показывает лучшую пропускную способность независимо от увеличения количества тайлов. На втором месте по пропускной способности находится ГИС-сервер

Mapnik/TileMill. ГИС-сервер MapServer показывает стабильное время загрузки на большом количестве тайлов, в то время как при загрузке малого количества тайлов его показатели являются одними из самых худших, также как и у ГИС-сервера Map Suite. Что касаетя ГИС-сервера GeoServer, то его показатели оказались самыми худшими, особенно при его запуске на платформе JRE8.

А.2 Тестирование загрузки ГИС-серверами тайловых слоев с перепроецированием

В этом разделе проверяется скорость перепроецирования векторных данных ГИС-серверами. Задача состоит в том, чтобы перепроектировать координаты из проекции WGS84 (EPSG: 4326) в сферическую проекцию Меркатора (EPSG: 3857). Результат представлен на рисунке А.3.

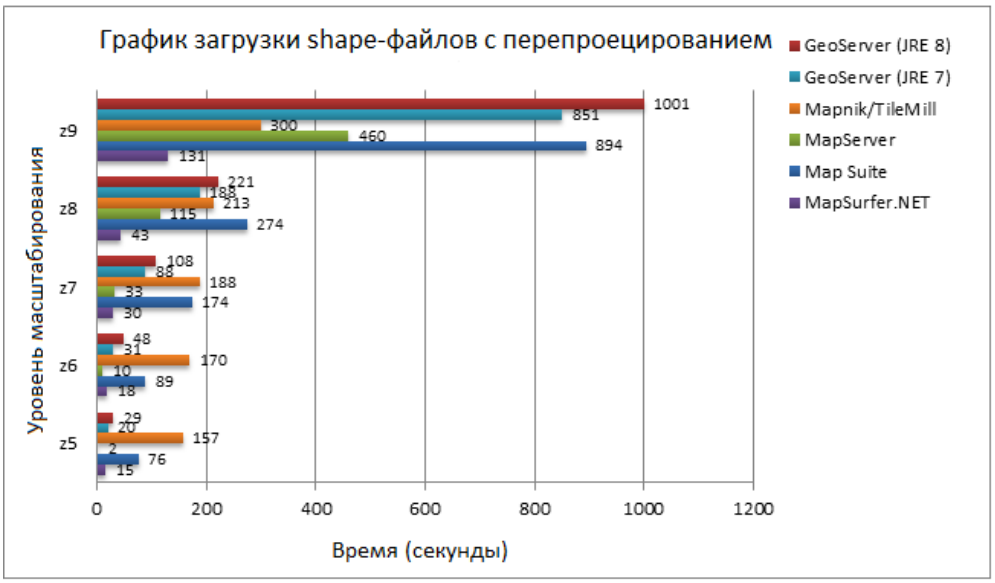

Рисунок А.3 – Результат запуска рендеринга shape-файла с перепроецированием и тайловой подгрузкой, где ось абсцисс показывает время, которое потребовалось для рендеринга, а ось ординат – масштаб

При загрузке shape-файлов с перепроецированием ГИС-сервер GeoServer, запущенный под JRE8, показывает наихудший показатель загрузки тайлов при

высоком коэффициенте масштабирования, в то время как при низком уровне масштабирования его показатели находятся на среднем уровне. ГИС-сервер Map Suite имеет показатели чуть лучше, чем ГИС-сервер GeoServer в данном тестировании. Лучшие показатели имеет ГИС-сервер MapSurfer.NET. Примерно одинаковые показатели вне зависимости от увеличения уровня масштабирования имеет ГИС-сервер Mapnik/TileMill. ГИС-сервер MapServer показывает отличный результат при загрузке тайлов в малом масштабировании, в то время как при увеличении масштаба его показатели становятся средними.

А.3 Тестирование загрузки ГИС-серверами тайловых слоев из базы данных PostgreSQL (PostGIS) без перепроецирования

В этом разделе приведены результаты тестирования загрузки ГИС-серверами тайловых слоев из базы данных PostgreSQL (PostGIS) без перепроецирования. Shape-файл был импортирован в таблицу PostgreSQL с использованием проекции EPSG: 3857. На рисунке А.4 представлен график тестирования.

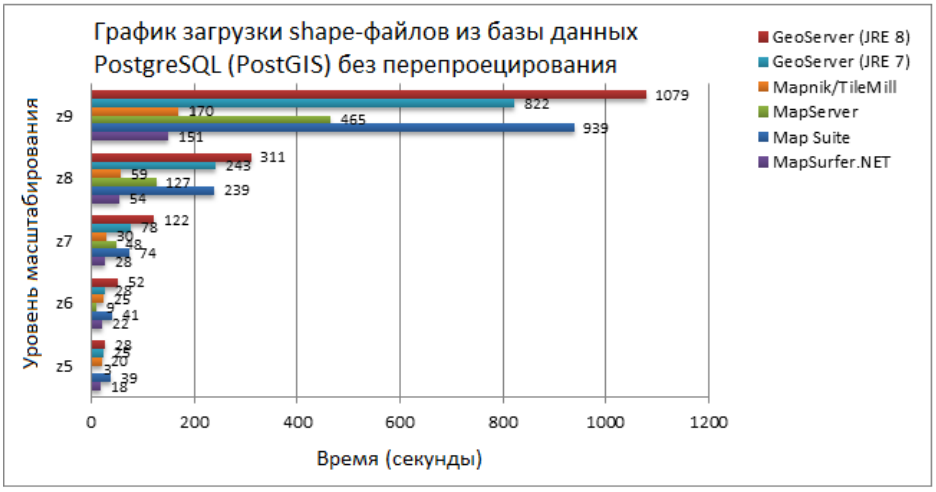

Рисунок А.4 – Результат запуска рендеринга shape-файла с тайловой подгрузкой из базы данных PostgreSQL (PostGIS) без перепроецирования, где ось абсцисс показывает время, которое потребовалось для рендеринга, а ось ординат – уровень масштабирования

Как следует из рисунка А.4, ГИС-серверы Mapnik/TileMill и MapSurfer.NET показывают лучшие показатели по результатам рендеринга shape-файла с тайловой подгрузкой из базы данных PostgreSQL (PostGIS) без перепроецирования. Худшие показатели имеет ГИС-сервер GeoServer, запущенный как под JRE8, так и под JRE7, хотя ГИС-сервер GeoServer ГИС-сервер GeoServer показывает более высокий результат, чем ГИС-сервер Map Suite. ГИС-сервер MapServer имеет отличные показатели при относительно небольшом масштабировании, в то время как при увеличении масштаба его показатели становятся средними.

А.4 Общие результаты тестирования ГИС-серверов

Общая сводка результатов тестирования представлена на рисунке А.5.

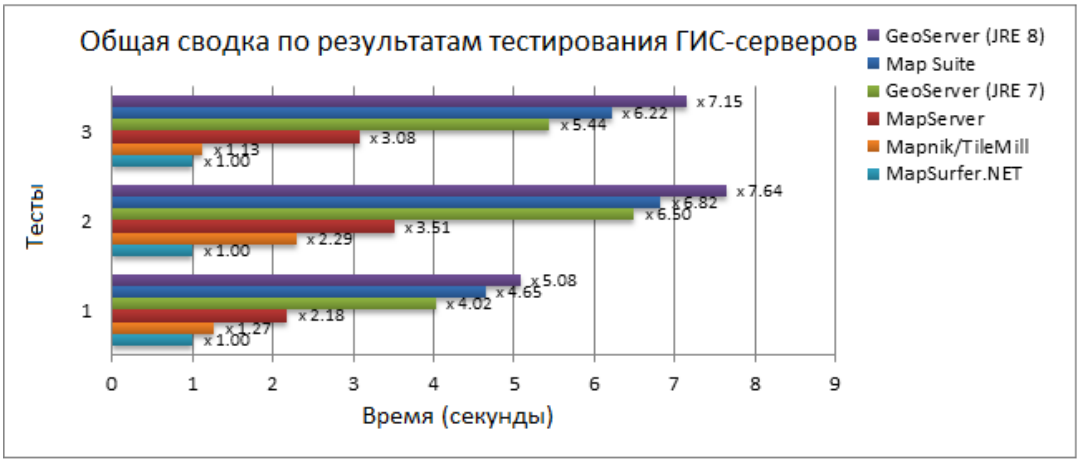

Рисунок А.5 – Общая сводка по результатам тестирования ГИС-серверов, где по оси абсцисс показано время загрузки shape-файла в секундах, а по оси ординат – номер теста

Судя по результатам, представленным в общем графике на рисунке А.5, наихудшую производительность показал ГИС-сервер Map Suite. Это можно объяснить тем, что Map Suite не поддерживает метатипирование и не поддерживает возможность сохранять 8-битные PNG-файлы. Тем не менее, в

относительно больших масштабах (z8, z9) он показывает лучшую производительность, чем ГИС-сервер GeoServer, запущенный под JRE 8. В то же время ГИС-сервер GeoServer с использованием JRE 7 работает быстрее, чем под JRE 8. ГИС-сервер GeoServer показывает впечатляющие результаты сравнительного анализа на небольших масштабах (z0- z4) в тесте №1. Это может быть объяснено тем, что он хранит весь shape-файл в памяти.

ПРИЛОЖЕНИЕ Б

Тестирование СУБД бенчмаркингами Jackpine

Б.1 Микро-бенчмаркинг СУБД

В таблице Б.1 показано время, затраченное СУБД на импорт shape-файлов, индексирование и обновление состояния СУБД.

Таблица Б.1 – Время загрузки shape-файлов СУБД

№ слоя	Размер данных, MB	Размер атрибутивной информации, MB	MySQL	PostgreSQL	Informix
1	0,89	0,96	5.786 сек	1.648 сек	31.9 сек
2	27,8	1,6	9.099 сек	2.355 сек	19.1 сек
3	196	25,5	4 мин 56.061 сек	1 мин 48.98 сек	20 мин 2.1 сек
4	1612	407,1	74 мин 25.482 сек	34 мин 42.839 сек	436 мин 53 сек
Сумма	1836,69	435,16	80 мин	37 мин	7.5 часов

Как следует из таблицы 1, самый быстрый показатель загрузки имеет СУБД PostgreSQL, в то время как СУБД MySQL имеет показатель, который в 2 раза хуже, чем у PostgreSQL. Загрузка данных в СУБД Informix при помощи утилиты loadshp имеет самый высокий показатель уровня загрузки, поскольку сервер по умолчанию проходит через контрольные точки каждые 1000 записей, что ведет к значительным задержкам. Значение в 1000 записей использовалось по умолчанию, при желании его можно изменить. Скорость loadshp может быть улучшена с помощью буферизации строк перед их записью СУБД. Однако это ограничивает возможность обработки ошибок, поскольку, если во время

загрузки возникает ошибка, все буферизованные строки с момента последней успешной записи будут потеряны.

Б.2 Микро-бенчмаркинг СУБД

В данной главе описаны результаты микро-бенчмаркинга СУБД. Его суть состояла в запуске заранее подготовленных запросов к СУБД, отражающих пространственные отношения между объектами. На рисунке Б.1 приведен один из результатов запуска микро-бенчмаркинга, осуществляющего запросы определения отношения одного объекта к другому.

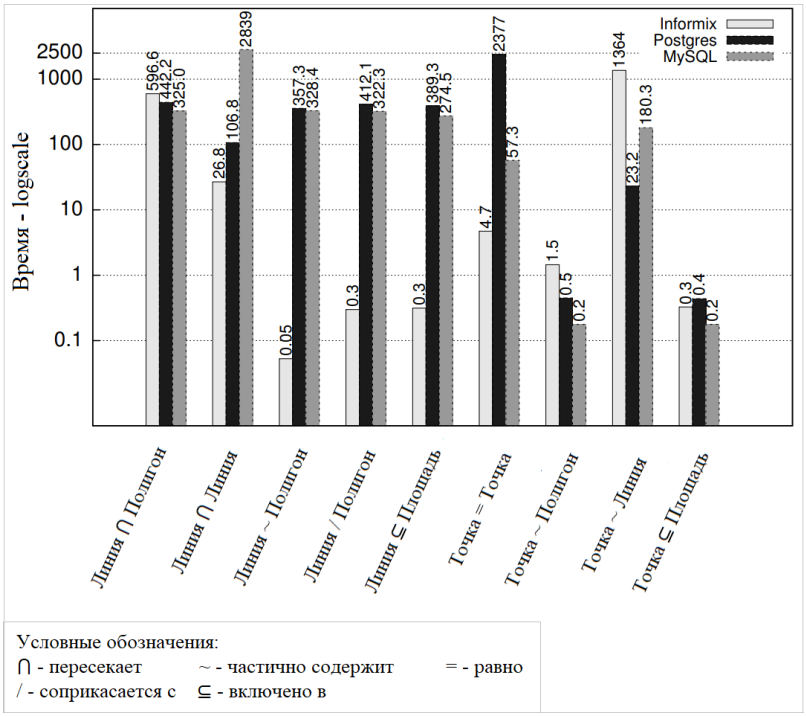

Рисунок Б.1 – Результаты тестирования по микро-бенчмаркингу Jackpine

Полученные результаты являются весьма неоднозначными. Некоторые запросы одна СУБД выполняет быстрее, чем другие СУБД, но при этом на другие запросы у той же СУБД затрачивается гораздо больше времени. Тем не менее, СУБД Informix справляется с большинством запросов быстрее, чем

Postgres и MySQL, которые в большинстве запросов показывают примерно одинаковое время выполнения.

Б.3 Макро-бенчмаркинг СУБД

В данной главе описаны результаты макро-бенчмаркинга СУБД.

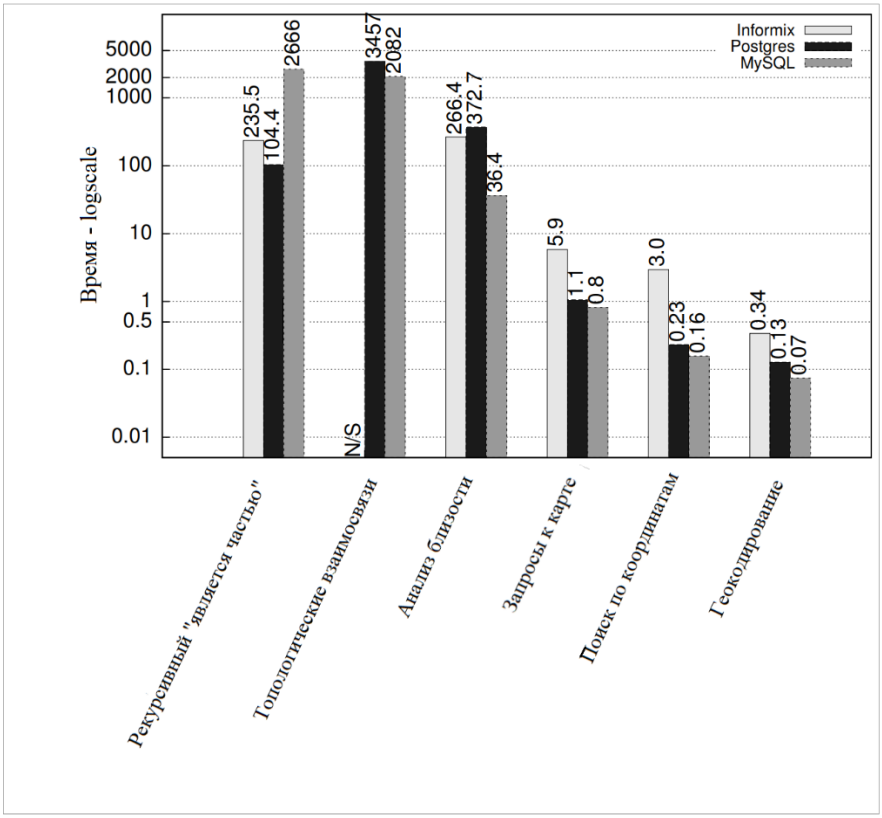

Рисунок Б.2 – Результаты тестирования по макро-бенчмаркингу Jackpine

Как видно из графика на рисунке Б.2, при геокодировании все 3 тестируемые СУБД показывают примерно одинаковый результат, но при этом СУБД MySQL потребовалось чуть меньше времени, чем двум другим. СУБД Informix показывает самый высокий результат по времязатратам в запросах поиска по координатам и запросам к карте на получение атрибутивной и другой информации. В тестировании скорости построения топологических связей использовались операции наложения и пересечения с использованием функции Dwithin (расстояние в пределах определенного смещения). Dwithin

поддерживается только в PostgreSQL. СУБД Informix не поддерживает выполнение данной функции напрямую, но для подобных операций в данной СУБД используется комбинация операций с функцией вычисления дистанции. Сработали СУБД PostgreSQL и MySQL примерно одинаково. В рекурсивном поиске отношений «является частью» между объектами лучший результат показала СУБД PostgreSQL.

ПРИЛОЖЕНИЕ В

Листинг программного кода инструментов мобильной ГИС

Листинг 1 – Программный код компонентов карты, иконок и кнопок

```
<MapView style={styles.map} // компонент из библиотеки react-native-map
        mapType={MAP_TYPES.STANDART} // подложка
        ref={ref => { this.map = ref; }} // ссылка на DOM-элемент карты
        initialRegion={{ // аттрибуты для показа на начальном экране
                latitude: 56.016698,
                longitude: 92.875955,
                latitudeDelta: LATITUDE_DELTA,
                longitudeDelta: LONGITUDE_DELTA,
        }}
        onPress={this.onPress} // функция-обработчик нажатия на карту
        >
    <Marker
        coordinate={{"latitude": x.Y,"longitude": x.X}} // координаты маркера
        image={medical} // иконка маркера
        onPress={() => this.selectPoint(x)} // обработчик нажатия на маркер
        style={styleForMedicals(x.X, x.Y, x.TYPE)}> // стиль оботражения маркера
    </Marker>
    <TouchableOpacity // область для нажатия (аналог кнопки в React Native)
        onPress={() => this.fitToMe()} // обработчик нажатия на кнопку
        style={[styles.bubble, styles.button]}> // стиль кнопки
        <Image // компонент картинки для отображения иконки в кнопке
                source={houseIcon}/> // иконка кнопки
    </TouchableOpacity>
</MapView>
```

Листинг 2 – Программный код построения зоны поиска

```
onPress(e) { // функция-обработчик нажатия на карту
    const { editing, creatingHole, painting, isModalOpen} = this.state;
    if (!painting) { return {}; } // если инструмент рисования не активен – ничего не делать
    if (!creatingHole) { // если рисование зоны поиска закончено
        this.setState({ //отправить в хранилище состояния проекта
            editing: {
                ...editing,
                coordinates: [ // координаты зоны поиска
                ...editing.coordinates,
```

```
                        e.nativeEvent.coordinate,
                    ],
                },
            });
        } else { // если рисование зоны поиска продолжается
            this.setState({ //отправить в хранилище состояния проекта
                editing: {
                    ...editing,
                    id: id++, // увеличение id для обновления интерфейса
                    coordinates: [ // координаты зоны поиска
                        ...editing.coordinates,
                    ],
                },
            });
        }
}
startPainting () { // функция-обработчик начала рисования зоны поиска
    this.setState({painting: true}) // флаг, показывающий что рисование начато
}
finish() { // функция-обработчик окончания рисования зоны поиска
    const { polygons, editing } = this.state;
    this.setState({ //отправить в хранилище состояния проекта
        polygons: [...polygons, editing], // id полигона
        painting: false, // флаг, показывающий что редактирование закончено
    });
    let bigLongitude = 180; // минимальные и максимальные значения широты и долготы
    let smallLongitude = -180;
    let bigLatitude = 90;
    let smallLatitude = -90;
    for (const i = 0; i < this.state.editing.coordinates.length; ++i) { // вычисление наибольшего
значения долготы координат, попавших в зону поиска
        if (this.state.editing.coordinates[i].longitude > bigLongitude) {
            bigLongitude = this.state.editing.coordinates[i].longitude;
        } else if (this.state.editing.coordinates[i].longitude < smallLongitude) { //
вычисление наименьшего значения долготы координат, попавших в зону поиска
            smallLongitude = this.state.editing.coordinates[i].longitude
        }

        if (this.state.editing.coordinates[i].latitude > bigLatitude) { // вычисление
наибольшего значения широты координат, попавших в зону поиска
            bigLatitude = this.state.editing.coordinates[i].latitude;
            } else if (this.state.editing.coordinates[i].latitude < smallLatitude) { //
вычисление наименьшего значения широты координат, попавших в зону поиска
                smallLatitude = this.state.editing.coordinates[i].latitude
            }
```

```
                }

                this.setState({bigLongitude, smallLongitude, bigLatitude, smallLatitude})
//отправить в хранилище состояния проекта
}
```

ПРИЛОЖЕНИЕ Г

Листинги программного кода инструментов веб-ГИС

Г.1 Листинг программного кода получения координат пользователя

```
const map = new ol.Map({ //создание объекта карты
	controls: ol.control.defaults({ /* добавление шкалы масштаба и координат мыши на карту
*/
		attributionOptions: {
		collapsible: false
	}
}).extend([mousePositionControl, scaleLineControl]),
	layers: [raster, vecLayer, vector, vectorMeasure], /* определение слоев на карте */
	target: 'map', // id тега, в который вставляется карта
	view: new ol.View({ /* настройка центра карты и уровня приближения при
	инициализации карты */
	center: ol.proj.fromLonLat([92.88, 56.04]),
	zoom: 12,
	})
});
const geolocation = new ol.Geolocation({ /* определяется объект геолокации с той же проекцией,
что и у карты */
	projection: map.getView().getProjection()
});
geolocation.setTracking(true); /* обновлять местоположение пользователя при его перемещении
*/
let positionFeature = new ol.Feature(); /* инициируется точечный объект местоположения
пользователя */
positionFeature.setStyle(new ol.style.Style({ // задаются стили объекта местоположения
пользователя
	image: new ol.style.Circle({
	radius: 6,
	fill: new ol.style.Fill({
		color: '#3399CC'
	}),
	stroke: new ol.style.Stroke({
		color: '#fff',
		width: 2
	})
	})
}));
geolocation.on('change:position', function () { /* функция обновления местоположения
пользователя на карте, если его фактическое местоположения изменилось */
```

```
        const coordinates = geolocation.getPosition();
        positionFeature.setGeometry(coordinates ? new ol.geom.Point(coordinates) : null);
        MY_COORDS = ol.proj.toLonLat(coordinates);
});
new ol.layer.Vector({ /* добавление объекта местоположения пользователя на карту */
        map: map,
        source: new ol.source.Vector({
        features: [positionFeature]
        })
});
```

Г.2 Листинг программного кода построения зоны поиска и создания информационной панели

```
<script src="https://cdn.rawgit.com/bjornharrtell/jsts/gh-pages/1.0.2/jsts.min.js"></script>
//библиотека поиска точек внутри зоны
const typeSelect = document.getElementById('type'); /* получить DOM-узел с формой зоны
поиска */
const draw, select;
const reader_wkt = new jsts.io.WKTReader() /* объект-определитель точек в зоне поиска*/
function addInteraction() { /*функция добавления инструмента построения зоны поиска */
        var value = typeSelect.value; /* получить форму зоны поиска */
        if (value !== 'None') { /* проверка, включен ли инструмент рисования зоны поиска */
                var geometryFunction, maxPoints;
                if (value === 'Square') { // условие для формы прямоугольника
                value = 'Circle';
                geometryFunction = ol.interaction.Draw.createRegularPolygon(4, 45 * (Math.PI /
                180)); // функция построения формы на карте
                } else if (value === 'Box') { //условие для формы квадрата
                value = 'LineString';
                maxPoints = 2; //максимальное количество точек
                geometryFunction = function (coordinates, geometry) { /* функция построения
                формы на карте */
                        if (!geometry) {
                        geometry = new ol.geom.Polygon(null);
                        }
                var start = coordinates[0];
                var end = coordinates[1];
                geometry.setCoordinates([
                        [start, [start[0], end[1]], end, [end[0], start[1]], start]]);
                return geometry;
                };
        }
```

```
draw = new ol.interaction.Draw({
        source: vectorMeasure.getSource(),
        type: /** @type {ol.geom.GeometryType} */ (value),
        geometryFunction: geometryFunction,
        maxPoints: maxPoints
});
map.addInteraction(draw); // добавить функцию рисования на карту
draw.on('drawend', // функция обработки окончания рисования
function (evt) {
        if (evt.feature.getGeometry().getType() == 'Circle') { { /* если форма зоны поиска
        круг – найти все вхождения в эту зону */
                var geom_select = evt.feature.getGeometry();
                var pt = reader_wkt.read('POINT (' + geom_select.getCenter().join(' ') + ')')
                var jsts_geom_select = pt.buffer(geom_select.getRadius());
        } else { //найти все вхождения в зону поиска при остальных формах
                var geom_select = evt.feature.getGeometry();
                var jsts_geom_select = parser.read(geom_select);
        }
        var layer_select_geometries = vecLayer /* получение всех точек слоя и выборка
        тех, что попали в зону поиска */
        .getSource().getFeatures().filter(function (el) {
if (jsts_geom_select.contains(parser.read(el.getGeometry()))) { /* обработка точек,
попавших в зону поиска */
        $(".info-panel").animate({ /* открытие боковой информационной панели */
                left: '0'
        });
        map.removeInteraction(draw); /* удаление вспомогательного слоя построения
        зоны поиска */
        const infoList = document.getElementById('info-list'); /* получение DOM-узла
        информационной панели */
        while (infoList.firstChild) { /* удалить информацию о предыдущих найденных
        объектах */
                infoList.removeChild(infoList.firstChild);
        }
        const clinicForList = document.createElement('li'); /* создать DOM-элемент
        списка для объекта */
        clinicForList.className =
        'mdl-list__item mdl-list__item--three-line mdl-list__item--full-height clinics-item'; /*
        задать имя класса для элемена списка для объекта */
        const {
                NAME, ADDRESS, PHONE, WEB_REG_ID, X, Y} = el.getProperties(); /*
        получение атрибутивной информации об объекте внутри зоны поиска */
        fetch(`https://maps.googleapis.com/maps/api/distancematrix/json?units=metric&orig
        ins=${MY_COORDS[1]},${MY_COORDS[0]}&destinations=${Y},${X}&key=AI
```

```
zaSyDCli8Ynh14VLGHiPeHVQ9TTvOih_gio-U`) /* получение расстояния от местоположения пользователя до объекта */
.then(x => x.json())
.then(x => x.rows[0].elements[0].distance.text)
.then(distance => {
clinicForList.innerHTML =`
        <span class="mdl-list__item-primary-content">
        <span>${NAME}</span>
        <span class="mdl-list__item-text-body">
        ${ADDRESS}
        <div class="point-distance-wrapper"><span class="point-
        distance">~${distance}</span></div>
        <div class="point-phone-wrapper"><i class="material-
        icons">phone</i>${PHONE}</div>
        <i class="point-tip-wrapper"> ${STARRED_CLINICS.get(el.getId()) &&
        STARRED_CLINICS.get(el.getId()).tip ?
        STARRED_CLINICS.get(el.getId()).tip : "}</i> /* генерация информации об объекте и ее вставка в информационную панель  */
        </span>
        </span>
        <span class="mdl-list__item-secondary-content">
        <button id="show-dialog" class="mdl-button mdl-button--icon mdl-js-button mdl-js-ripple-effect" style="margin-bottom: 16px;" data-
        upgraded=",MaterialButton,MaterialRipple">
        <i class="material-icons">share</i>
        <span class="mdl-button__ripple-container"><span class="mdl-ripple is-
        animating point-star-wrapper"></span></span></button>
        <div style="margin-bottom: 16px;"><a class="mdl-list__item-secondary-
        action ${STARRED_CLINICS.get(el.getId()) ? 'starred' : 'non-starred'}"
        id="star" data-text='${el.getId()}' href="#"><i class="material-
        icons">star</i></a></div>
        <a class="mdl-list__item-secondary-action" data-text='${el.getId()}'
        href="https://web-
        registratura.ru/index.php?open=system/record&step=2&mode=1&cid=${
        WEB_REG_ID}" target="_blank"><i class="material-
        icons">calendar_today</i></a>
        </span>`
})
.then(() => { /* изменение стиля найденных объектов и очистка вспомогательного объекта зоны рисования */
document.getElementById('info-list').appendChild(clinicForList);
vectorMeasure.getSource().clear();
select.getFeatures().clear();
select.getFeatures().extend(layer_select_geometries);
function delay_clear_layer() {
```

```
                timeoutID = window.setTimeout(clear_layer, 300);
                }
                function clear_layer() {
                        vectorMeasure.getSource().clear();
                }
                delay_clear_layer();
                });
                return true;
                }
                });
                },
                this
        );
        select = new ol.interaction.Select({ /* создание инструмента рисования зон поиска на
        карту */
                layers: [vecLayer],
                style: styles,
                condition: ol.events.condition.never
        });
        map.addInteraction(select); /* добавление инструмента рисования зон поиска на карту
        */
}
```

Г.3 Листинг программного кода построения тематической карты

```
$(document).on('click', '#starredColorPicker', function (e) { /* функция-обработчик изменения
цвета иконки для избранных медучреждений */
        STARRED_COLOR = $('#starredColorPicker').attr('src'); /* получение адреса выбранной
        иконки */
        $('.legend-body').append(`<img src='${STARRED_COLOR}'> - избранные
        медучреждения`);
        $('.legend').addClass('legend-big'); // изменение размеров легенды
        $('#starredColorPicker').addClass('starred-icon-checked');  /* изменение стиля кнопки с
        выбранной иконкой */
        $('.ol-scale-line').addClass('ol-scale-high'); //сдвиг шкалы масштаба
        const featuresProps = vecLayer.getSource().getFeatures().map(x => x.getProperties()); /*
        получение всех атрибутов объектов медучреждений */
        var feature, geometry;
        const newFeatures = [];
        for (let x of featuresProps) { /* перебор атрибутов объектов медучреждений */
                for (let c of STARRED_CLINICS.values()) { /* перебор списка избранных
                медучреждений */
                        if (c.name === x.NAME) {  /* проверка, является ли объект избранным */
```

```
                feature = new ol.Feature({ /* создание новой сущности */
                geometry: x.geometry
            });
            feature.setStyle(
            new ol.style.Style({
                image: new ol.style.Icon({
                anchor: [0.5, 1],
                src: STARRED_COLOR
            })
            })
        );
        newFeatures.push(feature); // добавление сущности в массив сущностей
        }
        }
        }
        var vectorSource = new ol.source.Vector({
            features: newFeatures
        });
        var vector = new ol.layer.Vector({
            source: vectorSource
        });
        map.addLayer(vector); /* добавление массива сущностей на карту */
});
```

Г.4 Листинг программного кода добавления заметки

```
$(document).on('change keyup paste', '#starredTextarea', function (e) { /* функция-обработчик изменения значения текстового поля для добавления заметки */
    const {id} = document.querySelector('#starredTextarea').dataset; /* получение id объекта, к которому будет добавлена заметка */
    const objById = STARRED_CLINICS.get(id); /* получение объекта по id из списка Избранных медучреждений */
    objById.tip = $('#starredTextarea').val(); /* получение значения текстового поля */
    STARRED_CLINICS.set(id, objById); /* добавление пары ключ-значение с полученной заметкой к объекту  */
});
```

Г.5 Листинг программного кода модального окна «Рассказать о медучреждении»

```
<script src="https://cdn.jsdelivr.net/npm/sharer.js@latest/sharer.min.js"></script> /* подключение библиотеки sharer.js для отправки сообщения в социальные сети и мессенджеры  */
<dialog class="mdl-dialog mdl-dialog_wide" id="dialog-recommend">
	<h4 class="mdl-dialog__title">Рассказать об этом месте?</h4> /* заголовок модального окна  */
	<div class="mdl-dialog__content">
		<p>Рекомендую обратить внимание на эту клинику!</p> /* информационная подпись к сообщению  */
		<img class="mdl-dialog__image" src="http://s1.iconbird.com/ico/1012/EcommerceBusinessIcons/w256h256135082312 8maps256x256.png"/> /* картинка-заглушка в модальном окне  */
	</div>
	<div class="mdl-dialog__actions"> /* кнопки отмены и социальных сетей  */
		<button type="button" class="mdl-button close">Отмена</button>
		<div class="mdl-dialog__content-icons">
		<i id="vk-recommend" class="sharer button fab fa-vk" data-sharer="vk" data-title="Рекомендую обратить внимание на эту клинику!" data-url="http://www.google.com/maps?q=56.235855,90.474075"></i> /* заголовок модального окна  */
		<i id="twitter-recommend" class="sharer button fab fa-twitter" data-sharer="twitter" data-title=" Рекомендую обратить внимание на эту клинику!" data-url="https://ellisonleao.github.io/sharer.js/"></i>
		<i id="facebook-recommend" class="sharer button fab fa-facebook" data-sharer="facebook" data-title=" Рекомендую обратить внимание на эту клинику!" data-url="https://ellisonleao.github.io/sharer.js/"></i>
		</div>
	</div>
</dialog>
```

Г.6 Листинг программного кода инструмента измерения расстояний

```
var source = new ol.source.Vector(); /* объект вспомогательной линии построения маршрута*/
var sketch; /* линейный геометрический объект маршрута*/
var helpTooltipElement; /* объект всплывающей подсказки*/
var helpTooltip;
var measureTooltipElement; /* объект всплывающей подсказки с расстоянием*/
var measureTooltip;
var continueLineMsg = 'Продолжить построение линии';
var pointerMoveHandler = function(evt) { /* функция-обработчик движения мыши по карте*/
```

```
if (evt.dragging) {
        return;
}
var helpMsg = 'Начать построение линии';
if (sketch) { /* если включен инструмент построения маршрута – получить геометрию, которая строится пользователем и сменить текст всплывающей подсказки*/
        var geom = (sketch.getGeometry());
        helpMsg = continueLineMsg;
}
helpTooltipElement.innerHTML = helpMsg;
helpTooltip.setPosition(evt.coordinate); /* обновить позицию всплывающей подсказки*/

helpTooltipElement.classList.remove('hidden');
map.on('pointermove', pointerMoveHandler); /* добавить функцию рисования на карту*/
map.getViewport().addEventListener('mouseout', function() { /*функция-слушатель нажатия кнопки мыши, скрывающая информационную подсказку*/
        helpTooltipElement.classList.add('hidden');
});
var draw;
function addInteraction() { /* функция обработки взаимодействия с картой*/
        draw = new ol.interaction.Draw({ /* создание линейного объекта рисования маршрута и его стилизация*/
                source: source,
                type: type,
                style: new ol.style.Style({
                fill: new ol.style.Fill({
                color: 'rgba(255, 255, 255, 0.2)'
                }),
                stroke: new ol.style.Stroke({
                color: 'rgba(0, 0, 0, 0.5)',
                lineDash: [10, 10],
                width: 2
                }),
                image: new ol.style.Circle({
                radius: 5,
                stroke: new ol.style.Stroke({
                color: 'rgba(0, 0, 0, 0.7)'
                }),
                fill: new ol.style.Fill({
                color: 'rgba(255, 255, 255, 0.2)'
                })
                })
                })
        });
```

```
map.addInteraction(draw); /* добавление функции обработки взаимодействия с картой на карту*/
createMeasureTooltip();
createHelpTooltip();
var listener;
draw.on('drawstart', /* функция, определяющая в каких единицах показывать расстояние – метрах или калометрах*/
	function(evt) {
		var formatLength = function(line) {
			var length = ol.Sphere.getLength(line);
			var output;
			if (length > 100) {
			output = (Math.round(length / 1000 * 100) / 100) +' ' + 'km';
		} else {
			output = (Math.round(length * 100) / 100) +' ' + 'm';
		}
	return output;
};
sketch = evt.feature;
var tooltipCoord = evt.coordinate;
listener = sketch.getGeometry().on('change', function(evt) { /*функция-обработчик добавления нового узла в маршруте*/
	var geom = evt.target;
	var output;
	output = formatLength(geom);
	tooltipCoord = geom.getLastCoordinate();
	measureTooltipElement.innerHTML = output;
	measureTooltip.setPosition(tooltipCoord);
	});
}, this);
draw.on('drawend', /* функция-обработчик окончания рисования: задание стилей информационной подсказки */
	function() {
		measureTooltipElement.className = 'tooltip tooltip-static';
		measureTooltip.setOffset([0, -7]);
		sketch = null;
		measureTooltipElement = null;
		createMeasureTooltip();
		ol.Observable.unByKey(listener);
	}, this);
}
function createHelpTooltip() { /* функция создания информационной подсказки: добавление на карту и удаление с карты */
	if (helpTooltipElement) {
		helpTooltipElement.parentNode.removeChild(helpTooltipElement);
```

```
        }
        helpTooltipElement = document.createElement('div');
        helpTooltipElement.className = 'tooltip hidden';
        helpTooltip = new ol.Overlay({
                element: helpTooltipElement,
                offset: [15, 0],
                positioning: 'center-left'
        });
        map.addOverlay(helpTooltip);
}
function createMeasureTooltip() { /* функция создания подсказки с измеренным расстоянием: добавление на карту и удаление с карты */
        if (measureTooltipElement) {
                measureTooltipElement.parentNode.removeChild(measureTooltipElement);
        }
        measureTooltipElement = document.createElement('div');
        measureTooltipElement.className = 'tooltip tooltip-measure';
        measureTooltip = new ol.Overlay({
                element: measureTooltipElement,
                offset: [0, -15],
                positioning: 'bottom-center'
        });
        map.addOverlay(measureTooltip)}
```

ПРИЛОЖЕНИЕ Д

Таблица сравнения характеристик ГИС-сервисов

Таблица Д1 – Сравнение характеристик ГИС-сервисов

ГИС-сервис	Вид	OSGEO	OGC-протоколы	Тайлы	Мета-данные	Язык API	Размер МБ
OpenLayers	Библиотека	Да	WMS,WFS	Да	Да	JavaScript	3,07
Leaflet	Библиотека	Нет	WMS	Да	Нет	JavaScript	1,43
GWT-OpenLayers	Обёртка библиотеки	Нет	WMS, WFS	Да	Да	Java	6,97
OL4JSF	Обёртка библиотеки	Нет	WMS, WFS	Да	Да	Java, JavaScript	0,02
ReadyMap SDK	Инструментарий	Нет	WMS	Да	Да	JavaScript	76
GeoExt	Инструментарий	Нет	WMS, WFS, WFS-T	Да	Да	JavaScript	50
Mapbender	Фреймворк	Да	WMS, WFS, WFS-T, CSW	Да	Да	PHP	95
CartoWeb	Фреймворк	Нет	WMS, WFS	Нет	Нет	PHP	52

Окончание приложения Д.

ГИС-сервис	Вид	OSGEO	OGC-протоколы	Тайлы	Мета-данные	Язык API	Размер МБ
MapFish	Фреймворк	Да	WMS, WFS	Да	Да	Java JavaScript PHP Python Ruby	107
QGIS Web Client	Клиентское приложение	Да	WMS	Да	Да	–	16
Geoide	Клиентское приложение	Нет	WMS, WFS, WFS-T	Да	Да	Action-Script 2 JavaScript	78
msCross	Клиентское приложение	Нет	WMS, WFS	Нет	Нет	JavaScript	–
Яндекс Конструктор карт	Клиентское приложение	Нет	WMS,WFS	Да	Да	JavaScript	–