МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

*Формирование карты рейтинга жилых районов*

Курсовая работа

09.03.04 Программная инженерия

Зав. кафедрой д. т. н., профессор М.Г. Матвеев

Обучающийся А.В. Калиткин

Руководитель Н.К. Самойлов

Воронеж 2021

Содержание

[Содержание 2](#_Toc73579042)

[Введение 3](#_Toc73579043)

[1 Постановка задачи 4](#_Toc73579044)

[2 Разработка приложения 5](#_Toc73579045)

[2.1 Преимущества и недостатки применения DHT 5](#_Toc73579046)

[2.2 Терминология 6](#_Toc73579047)

[2.3 Применение распределённой хэш-таблицы в Bittorrent 6](#_Toc73579048)

[2.4 Основы работы клиента сети Mainline DHT 7](#_Toc73579049)

[2.5 Таблица маршрутизации узла 9](#_Toc73579050)

[2.6 Использование протокола Mainline DHT в собственном приложении 10](#_Toc73579051)

[2.7 Особенности работы приложения 12](#_Toc73579052)

[Заключение 14](#_Toc73579053)

[Список использованных источников 16](#_Toc73579054)

[Приложение А 17](#_Toc73579055)

Введение

Выбор подходящего места для покупки или аренды жилья – одна из задач, с которой сталкиваются люди при планировании или уже в процессе переезда на новое место жительства. Как правило, на выбор всегда оказывает влияние большое количество факторов. К ним можно отнести ограниченность предложения на рынке, отсутствие вариантов с удобным расположением и приемлемой ценой, наличие поблизости социальной инфраструктуры и остановок общественного транспорта, уровень преступности в данном районе, экологическую обстановку и многие другие.

В случае, если возможных вариантов всего несколько, каждый из них можно досконально изучить самостоятельно. В том случае, если выбор достаточно велик или будущее место жительства не привязано к конкретному району города, на самостоятельное изучение обстановки придётся потратить во много раз больше времени.

Один из способов упросить анализ – использовать некую метрику для оценки всех возможных вариантов. Эта метрика может быть как субъективной и формироваться на основе данных социологических опросов, так и объективной и основываться на данных из открытых источников. Но в любом случае, единого варианта существовать не может, так как каждый человек ведёт свой образ жизни и имеет разные потребности, по-разному оценивает важность тех или иных критериев.

Критерий, который очень важен проживающим в городе семьям – наличие поблизости инфраструктуры и возможность удобно добираться из дома до неё и до места учёбы или работы всех членов семьи. Таким образом, метрика, вычисляемая на основе этого критерия будет подходить большинству, кто столкнётся с необходимостью оценить качество того или иного места в городе с точки зрения комфортного проживания.

Для удобства восприятия, полученную метрику логично будет представлять в виде тепловой карты, отображаемой поверх карты города.

1. Постановка задачи

Цель данной работы – разработать приложение, позволяющее пользователям строить тепловую карту качества района на основе данных об инфраструктуре из открытых источников.

Конечные пользователи – люди, изучающие другой город с целью выбора наилучшего возможного будущего места жительства в связи с планируемым переездом.

Разрабатываемое приложение должно соответствовать следующим требованиям:

* Не должно требовать установки
* Использовать только открытые данные
* Должно быть интерактивным с примитивным интерфейсом
* Не должно требовать от пользователя специальных знаний

Для выполнения поставленной задачи необходимо:

* Провести анализ предметной области
* Определить преимущества и недостатки существующих решений
* Выбрать источник открытых данных об инфраструктуре
* Выбрать средства реализации и архитектуру приложения
* Разработать приложение

1. Анализ предметной области
   1. Обзор существующих решений

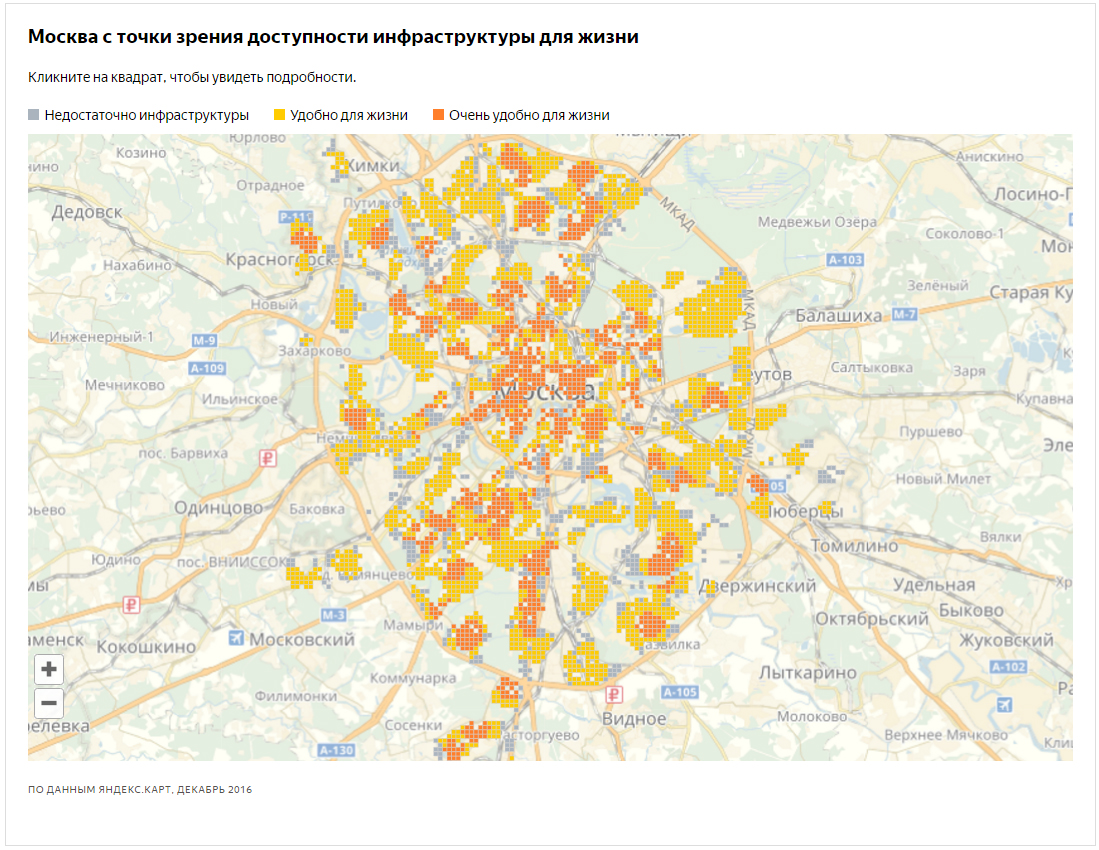
В ходе анализа предметной области было выяснено, что в большинстве случаев рейтинги районов городов составляются по результатам опросов общественного мнения, в которых жителей просят оценить их текущее место проживания в городе по нескольким критериям. Примерно половину из этих критериев составляют вопросы об обстановке в районе в целом с упором на такие аспекты, как чистота, экология, безопасность и работа коммунальных служб; остальные – о различных видах инфраструктуры: магазинах, развлечениях, инфраструктуры для детей, общественном транспорте и т. д.



1. Рейтинг районов Воронежа по результатам опросов жителей

Такие опросы позволяют оценить в целом район города, получить некоторую цифру, по которой этот район можно сравнить с другими, а также выявлять тенденции к изменению тех или иных показателей, но они имеют недостаточно высокую точность. Для того, чтобы привязка полученного рейтинга к географическому положению жителей была максимально точной, а погрешность из-за субъективности в ответах на опросы была минимальной, необходимо проводить опрос, имея довольно большую выборку, что определяет большую сложность данной задачи.

Из общего ряда выделяются исследования Яндекса, посвящённые Москве и Санкт-Петербургу с точки зрения доступности инфраструктуры для жизни. Эти исследования были проведены аналитически, используя данные Яндекс.Карт об организациях. В ходе исследования наиболее заселённые районы городов были разделены на квадраты со стороной 300 метров, и затем каждый из квадратов относили к одному из трёх классов – «недостаточно инфраструктуры», «Удобно для жизни» или «Очень удобно для жизни» на основании доступности для жителей этого квадрата всей необходимой для жизни инфраструктуры в пешей доступности.



1. Проведённое Яндексом исследование Москвы с точки зрения доступности инфраструктуры для жизни

Аналогичные карты были составлены для инфраструктуры для развлечений, или учитывая все виды инфраструктуры вместе. Затем полученные по квадратам данные были сгруппированы по районам, к которым они относятся, и по полученным результатам были сформированы интерактивная карта и таблица с рейтингами и особенностями районов – относительной долей квадратов каждого из классов, среднее время пешком до заведений и организаций разных видов и комментарии исследователей.

Стоит так же упомянуть сервис Яндекс.Недвижимость, который кроме отображения на карте доступного для продажи или аренды жилья даёт так же даёт возможность посмотреть их собственный рейтинг инфраструктуры в виде тепловой карты. Но, к сожалению, он доступен только для Москвы и Санкт-Петербурга.

* 1. Выбор источника данных об инфраструктуры

На данный момент есть несколько сервисов, предоставляющих доступ к картам. Большое популярностью в России пользуются Яндекс.Карты и 2GIS, на западе – Google Maps. Однако это всё закрытые проекты, которые требуют регистрации и соблюдения некоторых ограничений даже в случае некоммерческого использования и не подходят требования проекта.

В качестве источника данных об инфраструктуре был выбран проект OpenStreetMap, предоставляющий большое количество разнообразных географических данных, которые могут быть использованы совершенно свободно.

Наполнение карты (количество обозначенных объектов, актуальность, количество метаданных), а так же её качество (соответствие метаданных принятым стандартам) может варьироваться от города к городу, но в большинстве городов России и почти всех городах Европы достаточно хорошее, чтобы рассчитывать свой собственный рейтинг на основе этих данных.

1. Реализация
   1. Выбор средства доступа к данным

Основу приложения должна составлять интерактивная карта, позволяющая выбрать на ней некоторый интересующий пользователя участок, на котором в последствии будет просчитаны рейтинги района и отрисована тепловая карта. Но для того, чтобы мы могли провести какие-то вычисления, необходимо сначала получить необходимые для этого данные.

Есть два пути решения этой проблемы. Первый – развернуть свой собственный сервер базы данных, который будет содержать данные того региона, с которым мы работаем. Один из плюсов данного подхода – возможность отправлять запросы на выборку данных или производить поиск маршрутов, обращаясь напрямую к серверу базы данных. Однако, объём данных OSM в несжатом виде со всеми индексами достаточно велик, что потребует наличия на сервере соответствующего количества места на жёстких дисках и довольно много времени для выполнения импорта и предобработки данных. Второй существенный минус – необходимость регулярного обновления базы данных для поддержания данных в актуальном состоянии, что расточительно при небольшом количестве пользователей системы.

Второй путь – не разворачивать свой сервер, а использовать открытые API. OpenStreetMap предоставляет два API – первый, editing API, используется только приложениями для редактирования карт, так как позволяет извлекать самые актуальные данные и вносить изменения на сервер. Он не позволяет извлекать данные с больших территорий и довольно ограничен. Второй, Overpass API, позволяет писать в декларативном стиле довольно сложные запросы к базе и извлекать с их помощью данные сразу на большой территории. Стоит заметить, что несмотря на свои возможности, этот API позволяет только извлекать объекты и проводить с ними манипуляции, но проводить с ними дополнительные вычисления.

В своём приложении для извлечения необходимых данных мы будем обращаться к одному из общедоступных серверов Overpass API. По заверению их администраторов, они достаточно производительны, чтобы их можно быть использовать, совершенно не волнуясь, что мы можем помешать работе других приложений, если наше приложение совершает в сутки не более 10 000 запросов и выгружает в сутки не более 10 ГБ данных.

Для тестирования запросов к Overpass API я использовал сервис Overpass Turbo, позволяющий писать и выполнять запросы, а так же отображать результат запроса в текстовом виде и на карте прямо в окне браузера.

* 1. Отрисовка тайлов

Аналогичная ситуация обстоит и с сервером для отрисовки тайлов – мелких кусочков изображения, из которых состоит отображаемая пользователю в растровом виде карта. Так как мы не имеем собственного сервера данных OSM и тайлового сервера, обращаться будем к общедоступному серверу MAPNIK, принадлежащего OSM.

* 1. Выбор технологий и архитектуры для приложения

Так как приложение не должно требовать от пользователей для своей работы его установки, логично реализовать его в виде одностраничного веб-приложения. Так же так как мы не имеем собственного сервера базы данных, а обращаемся за ними к общедоступным серверам, используя открытый API, у нас есть возможность производить все необходимые для расчёта рейтингов и отрисовки карты вычисления на стороне клиента, без необходимости иметь собственный сервер в принципе.

Для создания клиентского приложения я решил использовать фреймворк Vue и язык программирования TypeScript, для создания пользовательского интерфейса – библиотеку Vuetify, для отображения интерактивной карты – библиотеку Leaftet.

Заключение

улюлюлююлюлю

Список использованных источников

1. К. Нейгел, Б. Ивьен. Дж. Глинн, К. Уотсон, М. Скиннер: C# и платформа .NET 3.0 для профессионалов. Пер. с англ. – 2008
2. Мэтью Мак-дональд: WPF: Windows Presentation Foundation в .NET 3.0 для профессионалов. Пер. с англ. – 2008
3. Дж. Глейзер, С. Мадхав: Многопользовательские игры. Разработка сетевых приложений. – СПб.: Питер, 2017
4. Andrew Loewenstern, Arvid Norberg: DHT Protocol – 2008
5. Arvid Norberg: DHT routing table maintenance – 2014
6. Anil Can Akay: Distributed algorithms for improving BitTorrent performance – 2010
7. Liang Wang, Jussi Kangasharju: Measuring Large-Scale Distributed Systems – 2013
8. Liang Wang, Jussi Kangasharju: Real-World Sybil Attacks in BitTorrent Mainline DHT – 2013
9. Peter Maymounkov, David Mazieres, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric" – 2002

Приложение А

Программный код

//Методы установления соединения и передачи данных

//по протоколу UDP на языке C#

public void Connect()

{

any = new IPEndPoint(IPAddress.Any, port);

connection = new UdpClient(port);

connection.BeginReceive(new AsyncCallback(Receive), null);

}

public void Send(byte[] bytes, IPEndPoint endPoint)

{

connection.Send(bytes, bytes.Length, endPoint);

}

public void Send(BencodeDictionary query, IPEndPoint endPoint)

{

byte[] bytes = query.SaveToBytes();

connection.Send(bytes, bytes.Length, endPoint);

Console.WriteLine(Encoding.ASCII.GetString(bytes)); //Debug

}

public void Receive(IAsyncResult res)

{

byte[] received = connection.EndReceive(res, ref any);

ProcessMessage(received);

connection.BeginReceive(new AsyncCallback(Receive), null);

}