**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ПЕТРА ВЕЛИКОГО**

**Институт компьютерных наук и технологий**

**Кафедра «Распределенные вычисления и компьютерные сети»**

**КУРСОВОЙ ПРОЕКТ**

на тему: Построение оптимального маршрута

Выполнил студент гр. В.Б.Борисов

гр. 43507/1

Руководитель

доц. А.В.Самочадин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2016 г.

Санкт-Петербург

2016

**Оглавление**

[Введение 3](#_Toc470211871)

[1.1 Актуальность проблемы 3](#_Toc470211872)

[1.2 Постановка задачи 3](#_Toc470211873)

[Основная часть 4](#_Toc470211874)

[2.1 Механизм маршрутизации 4](#_Toc470211875)

[2.2 Выбор алгоритма построения маршрута 5](#_Toc470211876)

[3 Реализация приложения 8](#_Toc470211877)

[3.1 Выбор средств разработки 8](#_Toc470211878)

[3.2 Интерфейс и функциональные возможности 10](#_Toc470211879)

[3.2.1 Валидация приложения 10](#_Toc470211880)

[3.2.2 Панель управления 11](#_Toc470211881)

[Заключение 15](#_Toc470211882)

[Список литературы 16](#_Toc470211883)

# Введение

# 1.1 Актуальность проблемы

Данная тема без преувеличения является актуальной. Все мы хотя бы раз в жизни прокладывали маршрут, будь эта поездка в другой город или просто мы хотим попасть из одного района города в другой. У всех у нас возникает естественное желание достичь место назначения за минимальные затраты. Именно поэтому различные компании активно участвуют в решении этой задачи, улучшая старые алгоритмы или изобретая новые.

Всего каких-то десять-пятнадцать лет назад в бардачке каждого водителя лежал атлас дорог. Он и был главным помощником при планировании маршрута. Сейчас вместо атласа люди всё чаще используют электронные карты или мобильные приложения, и умные алгоритмы сами строят для них наилучший маршрут.

# 1.2 Постановка задачи

В рамках данной работы была поставлена задача проложить оптимальный маршрут и оценить время выполнения заказа курьера (который развозит почту). Надо оценить время поездки на машине от точки до точки плюс время от парковки до офиса внутри здания и обратно. При этом, считается что планы зданий есть. В результате должна быть получена оценка времени на выполнение всего заказа, с учетом всех передвижений.

# Основная часть

# 2.1 Механизм маршрутизации

Главные составляющие механизма маршрутизации — это дорожный граф и алгоритм, который рассчитывает путь.

Дорожный граф — это сетка дорог. Она состоит из множества фрагментов, которые состыкованы между собой. Например, дорожный граф Саратова (население — около 840 тысяч человек) состоит из 7592 фрагментов. Каждый из них несёт информацию о своём участке дороги: географические координаты, направление движения, средняя скорость, с которой машины обычно едут на этом участке, и другие параметры. Кроме того, каждый фрагмент содержит данные о том, как он стыкуется с соседними участками — есть ли в этом месте поворот направо или налево, можно ли там развернуться в обратную сторону или разрешается ехать только прямо.

Само собой, дорожный граф нельзя сделать раз и навсегда. Транспортная система города имеет обыкновение меняться. Появляются новые дороги и развязки, меняется направление движения. А там, где ещё недавно был поворот, может висеть «кирпич». Чтобы не отставать от жизни, такие именитые компании как Яндекс, Google, 2ГИС, между которыми есть здоровая конкуренция, регулярно обновляют данные.

Во-первых, постоянно обрабатываются сообщения о неточностях в графе, которые пользователи присылают с помощью мобильных приложений и веб-сервиса. С этими сообщениями работают эксперты, которые используют также открытые источники информации о транспортной системе (например, сайты местных администраций).

Во-вторых, для определения неточностей на карте дорог существует специальная система. Она фиксирует все случаи, когда данные о движении машин, которые анонимно передают водители, не совпадают с имеющейся сеткой дорог. Если это не случайный нарушитель, который выехал на газон или развернулся в неположенном месте, возможно, на этом участке изменилась схема движения. Все такие случаи разбираются, и при необходимости в граф вносятся изменения.

Дорожный граф хранится на серверах Яндекса в нескольких экземплярах — если какой-то из серверов будет временно недоступен, маршрутизация все равно будет работать.

# 2.2 Выбор алгоритма построения маршрута

Маршрут будем рассчитывать по алгоритму Дейкстры. С его помощью приложение будит вычислять самый быстрый вариант проезда — исходя из длины каждого отрезка графа и скорости движения на этом участке. Если пользователь строит маршрут проезда без учёта пробок, то алгоритм использует среднюю скорость движения на участке. А если пользователь хочет знать, как быстрее всего добраться до места с учётом ситуации на дороге, то алгоритм задействует данные о текущей ситуации на дороге.

Как это происходит, можно разобрать на примере. Представим, что нужно проложить маршрут из точки А в точку B. Алгоритм начинает методично перебирать все возможные варианты. Первым делом он прокладывает маршрут на один шаг (фрагмент графа) во все стороны от точки А. И затем вычисляет, сколько времени потребуется на преодоление этих участков (тут все просто — расстояние делится на скорость). Дальше он выбирает точку, до которой удалось бы добраться быстрее всего. Это точка С.

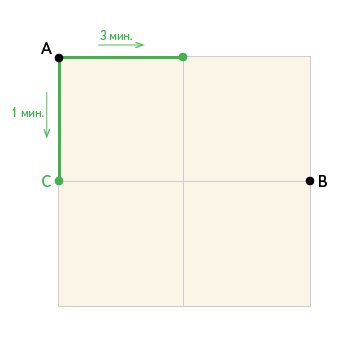


Рисунок 1

Затем алгоритм строит маршрут ещё на один шаг — во все стороны от точки С. И снова анализирует, в какую из точек можно было бы попасть быстрее всего. На этот раз это точка D. На следующем шаге алгоритм будет строить маршрут уже от неё.

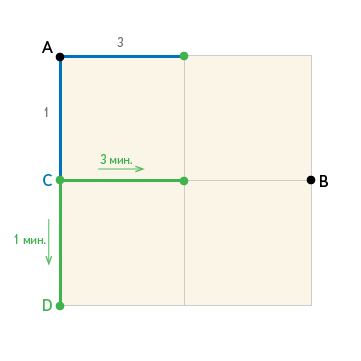


Рисунок 2

Продолжая в том же духе, маршрутизатор находит вариант проезда, который оказывается самым коротким по времени.

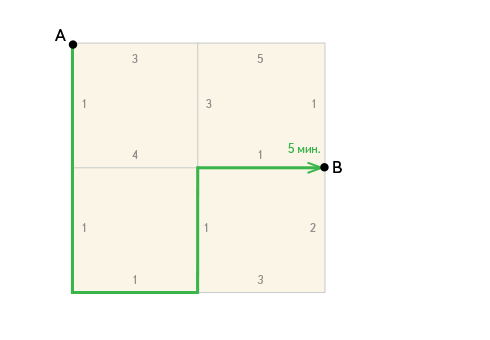


Рисунок 3

# 3 Реализация приложения

# 3.1 Выбор средств разработки

Для того чтобы решить данную задачу, а именно мы хотим проложить оптимальный маршрут и оценить весь путь курьера. Для начала нам понадобится карта, на которой и будем строить маршрут. На российском рынке картографических и справочных сервисов можно выделить трех основных игроков:

1. **Яндекс.Карты;**
2. **2ГИС;**
3. **Google Maps.**

Возникает вопрос: какой выбрать сервис? Я решил составить таблицу, в которой отметил, как мне кажется, главные критерии.

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии | Яндекс.Карты | 2ГИС | Google Maps |
| Покрытие | Лучшее покрытие России, уступает Google в покрытии мира | Уступает конкурентам в покрытии как в России, так и в других странах | **Лучшее покрытие всего мира** |
| Отображение пробок в крупных городах | **Да. Отображение доп. информации о дорожной обстановке** | Не все города | **Практически все города по всему миру** |
| Возможность загрузки и использования онлайн | Присутствует | Присутствует | Присутствует |
| Редактирования карт | Да | Да | Да |
| Построение маршрута | Автомобиль, общественный транспорт. Строит с учетом пробок. Требует интернет для построения | Автомобиль, общественный транспорт. Возможность отдельно выбрать вариант «Метро». Не требует интернета для построения маршрута | **Автомобиль, общественный транспорт, пешеходный маршрут. Возможность выбрать только один из видов транспорта или вариант пешком. Строит с учетом пробок и расписания общественного транспорта. Требует интернет для построения** |
| Документация | Хорошая, немного хуже чем у google maps | Хуже чем у Яндекс.Карт и Google maps | Наглядная и понятная |

После составления таблицы был сделан вывод, что для нашей задачи подходят все 3 картографических сервисов, поэтому можно выбрать любой. Для формирования и обработки запросов, создания интерактивного и независимого от браузера интерфейса был выбран язык javascript, потому что он используется в клиентской части веб-приложений: клиент-серверных программ, в котором клиентом является браузер, а сервером — веб-сервер, имеющих распределённую между сервером и клиентом логику. Обмен информацией в веб-приложениях происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами. Для реализации GUI были выбраны HTML и CSS.

# 3.2 Интерфейс и функциональные возможности

После запуска веб-приложения мы видим слева панель управления, с которой можно взаимодействовать, а остальную часть экрана занимает карта Санкт-Петербурга. (см. рис. 4)

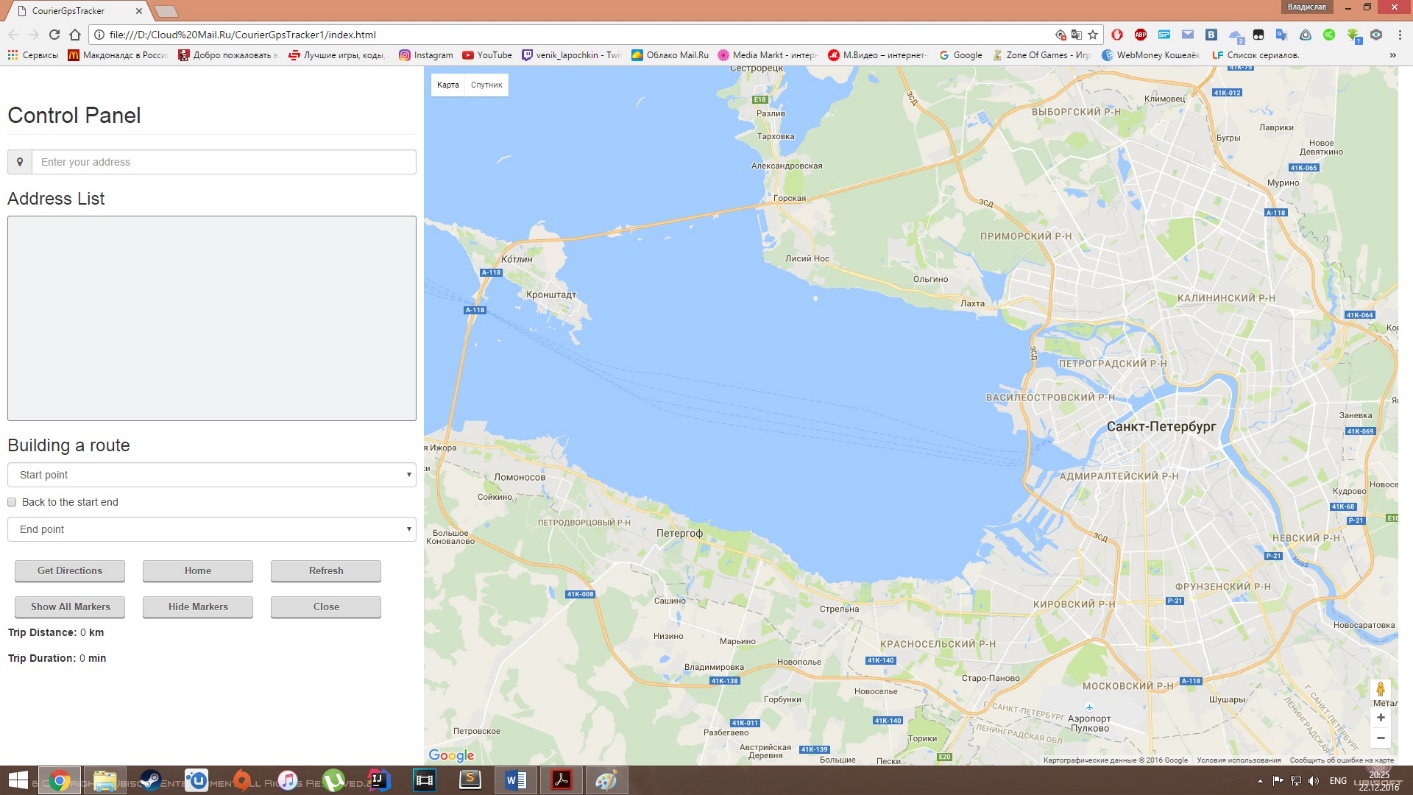


Рисунок 4

# 3.2.1 Валидация приложения

Перед непосредственно запуском приложения, стоит сказать при каких сценариях приложение не будет работать. (см. рис. 5)

Здесь реализована простейшая валидация. Приложение не построит маршрут если:

* В списке адресов меньше двух мест
* Не указана начальная и конечная точки
* Адрес, который вы ввели не был найден на карте. Чтобы эта ошибка не возникала, было принято решение использовать строку быстрой навигации. (см. рис. 6)

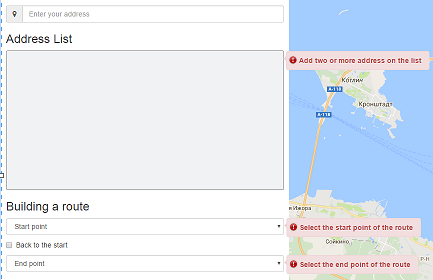


Рисунок 5

# 3.2.2 Панель управления

На панели управления мы видим поле ввода адреса. (см. рис. 6)

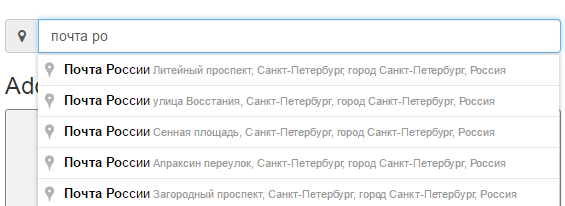


Рисунок 6

В нем мы пишем все интересующиеся нас адреса (например, начальный и конечный) и заносим в список адресов. После этого мы указываем (см. рис 7 и 8) начальный и конечный адрес.

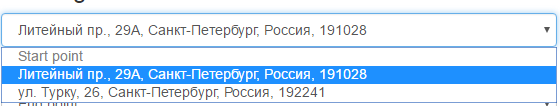


Рисунок 7

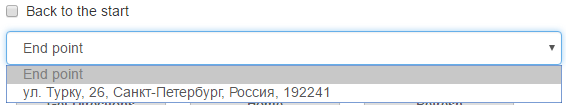


Рисунок 8

При этом если мы хотим проложить не просто маршрут из точки А в точку В, а еще и хотим вернуться назад, посчитав все расстояние и путь, то нужно поставить галочку напротив «Back to the start».

Заполнив все необходимые поля, мы нажимаем кнопку «Get directions» и на карте появляется оптимальный маршрут. (см. рис. 9)

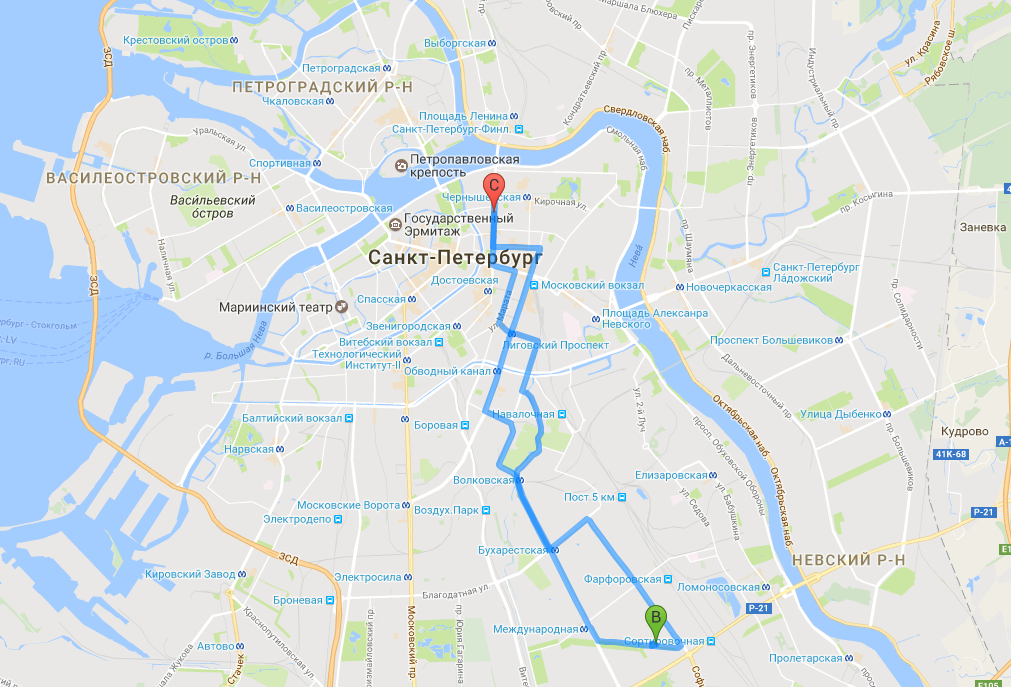
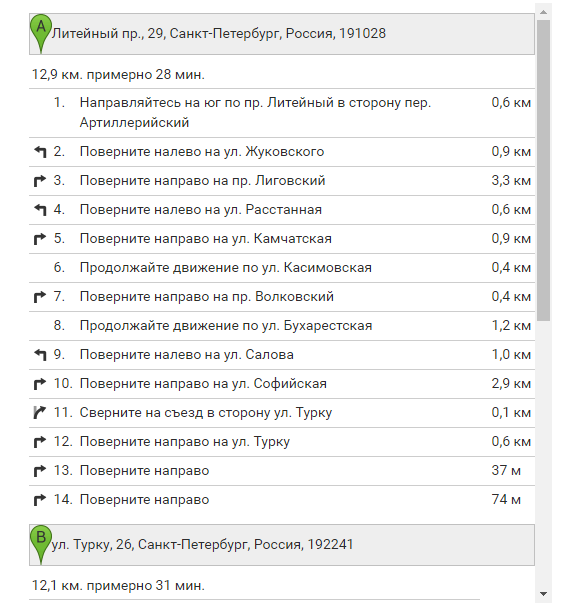


Рисунок 9

Помимо прокладки маршрута, появляется еще и таблица всех передвижений курьера, где написан адрес, поворот и расстояние. (см. рис. 10)



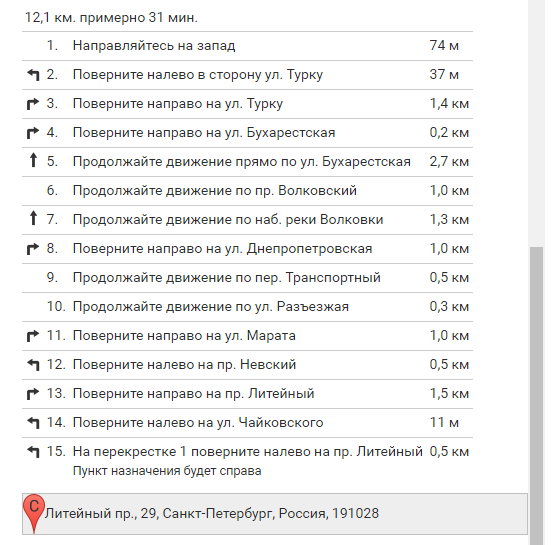


Рисунок 10

И как результат получаем весь пройденный путь и все время. (см. рис. 11)



Рисунок 11

# Заключение

В результате выполнения научно-исследовательской работы было реализовано web-приложение с полным функционалом. Данное приложение может не только прокладывать маршрут по всему городу, но и выводить таблицу всех передвижений пользователя, с последующим подсчетом всего пройденного пути и времени.

# Список литературы

1. Google Maps JavaScript API. - [Электронный ресурс]. – Режим доступа:https://developers.google.com/maps/documentation/javascript/tutorial?hl=ru
2. Причины ошибок в системе GPS // Paraclub.ru. - [Электронный ресурс]. – Режим доступа: http://www.paraclub.ru/NB/pogrshnost\_GPS.shtml
3. Stack Overflow. - [Электронный ресурс]. – Режим доступа:http://ru.stackoverflow.com/questions/tagged/google-maps-api
4. Google Maps API. - [Электронный ресурс]. – Режим доступа:https://habrahabr.ru/post/110460/
5. Решение задачи коммивояжера с привязкой к географическим координатам. - [Электронный ресурс]. – Режим доступа: http://www.market-journal.com/ekoproces/42.html
6. Задача коммивояжера. - [Электронный ресурс]. – Режим доступа: http://www.math.nsc.ru/LBRT/k5/OR-MMF/TSPr.pdf
7. Маршрутизация. - [Электронный ресурс]. – Режим доступа: https://yandex.ru/company/technologies/routes/