

УДК 004.031.4

**СОЗДАНИЕ МОБИЛЬНОГО
ПРИЛОЖЕНИЯ ДЛЯ СЕТИ
АВТОЗАПРАВОЧНЫХ СТАНЦИЙ***Д. В. Щербина, Н. Ю. Богданова*

Донской государственный технический
университет, г. Ростов-на-Дону,
Российская Федерация

Denis_dstu@mail.runatalyabogdanova@yandex.ru

Представлен проект создания мобильного приложения, функционал которого позволит находить ближайшую автозаправочную станцию к местоположению мобильного устройства, строить оптимальный маршрут от водителя к АЗС, получать информацию о АЗС. Приложение создается с помощью framework Ionic . В разработке будут применены некоторые директивы framework Angular.js

Ключевые слова: АЗС, search, ionic framework Angular.js, Google maps.

Введение. Понимание того, что человечество может сделать свою деятельность лучше, удобней и качественней, способствует активной мыслительной и творческой деятельности. В результате этой деятельности появляются новые совершенные технологии, способные значительно улучшить качество жизни.

Человек не только создаёт различные приборы для качественного, лёгкого и результативного труда, но также автоматизирует многие виды работ, освобождая людям время, а время - главный движущий элемент эволюции общества, его перехода от одной формы организации общественного устройства к другой. Именно время играет роль испытателя человеческих взаимоотношений, жизни вокруг индивида и в рамках общества.

Фреймворк — структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Ionic framework — один из самых широко обсуждаемых фреймворков — это SDK для создания гибридных мобильных приложений. SDK - Набор средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ. Следующие компоненты используются в ionic framework: набор CSS и JS компонентов, созданный на основе AngularJS, SASS (преобразователь css кода), Apache Cordova (мобильная среда разработки). Другими словами, это разработка мобильных приложений с использованием веб-технологий.

Постановка задачи. В рамках данной работы будет создано мобильное приложения на базе ИТ отдела компании ООО «Мегаполис», лидера на рынке АЗС (авто заправочные станции) Ростовской области. Разработка осуществляется инструментами ionic [1] и angular [2] фреймворков. Целью является определение ближайшей АЗС к местоположению устройства и построение маршрута к нему. Будет сделан удобный поиск по списку АЗС, создана история

UDC 004.031.4

**CREATION OF A MOBILE APPLICATION
FOR A FILLING STATION CHAIN***D. V. Shcherbina, N. Y. Bogdanova*

Don State Technical University, Rostov-on-Don,
Russian Federation

Denis_dstu@mail.runatalyabogdanova@yandex.ru

The paper presents the project for creation of a mobile application, the functionality of which will make it possible to find the nearest gasoline station to the location of a mobile device, to build the optimal route from a driver to the gasoline station and to receive information about the station. The application is written with the help of framework Ionic. Some framework Angular.js directives will be applied in the project development.

Key words: gasoline station, search, ionic framework Angular.js, Google maps.

поиска выбранных ранее АЗС и доступный вывод на экран всех АЗС в вашем городе.

Основная часть. Мобильное приложение — программа, разработанная с использованием фирменного стиля, которую клиенты самостоятельно устанавливают на свое мобильное устройство, чтобы иметь возможность быстрого доступа к необходимой информации. Такое приложение позволит клиентам пользоваться услугами, предоставляемыми компанией, что будет выгодно как для клиента, так и для заказчика.

Уровень автомобилизации России в 2016 г. составляет 285 машин на 1000 человек [3].

Первый этап создания приложения: сбор и анализ требований, который проводился стандартным способом: интервьюированием.

Прежде, чем начать написание программного продукта, необходимо изучить требования к безопасности рабочего места [4], после чего нужно развернуть необходимую среду. Для проекта требуются следующие программные пакеты:

- Android Studio;
- Java;

Рассмотрим каталог, содержащий проект (рисунок 1):

Имя	Дата изменения	Тип	Размер
hooks	28.01.2017 20:26	Папка с файлами	
platforms	28.01.2017 20:26	Папка с файлами	
plugins	28.01.2017 20:27	Папка с файлами	
resources	28.01.2017 20:27	Папка с файлами	
scss	28.01.2017 20:27	Папка с файлами	
www	28.01.2017 20:27	Папка с файлами	
.bowerrc	26.06.2016 20:01	Файл "BOWERRC"	1 КБ
.editorconfig	26.06.2016 20:01	Файл "EDITORCONFIG"	1 КБ
.gitignore	26.06.2016 20:01	Файл "GITIGNORE"	1 КБ
bower.json	26.06.2016 20:01	Файл "JSON"	1 КБ
config.xml	26.06.2016 20:50	Документ XML	5 КБ
gulpfile.js	26.06.2016 20:01	JetBrains WebStorm	2 КБ
ionic.project	26.06.2016 20:02	Файл "PROJECT"	1 КБ

Рис. 1 – структура документов.

На этом месте расположены все важные компоненты фреймворка, а в папке «www» размещен программный продукт. Раскроем папку (рисунок 2).

Имя	Дата изменения	Тип	Размер
.idea	28.01.2017 20:27	Папка с файлами	
css	28.01.2017 20:27	Папка с файлами	
images	28.01.2017 20:27	Папка с файлами	
img	28.01.2017 20:27	Папка с файлами	
js	28.01.2017 20:27	Папка с файлами	
lib	28.01.2017 20:27	Папка с файлами	
templates	28.01.2017 20:27	Папка с файлами	
index.html	22.06.2016 21:17	Chrome HTML Do...	2 КБ
points.json	29.05.2016 14:38	Файл "JSON"	400 КБ

Рис. 2 — Содержимое папки «www».

В папке «css» находятся стили создания данного проекта, в «img» — графические изображения, в «js» — все необходимые для работы приложения «js» файлы, в «lib» находятся файлы «ionic» и «angular» фреймворков.

HTML файлы, автоматически подгружаются из папки «templates» и динамично работают при запуске Web-приложения.

```
Denis@Denis-PC MINGW64 ~/Desktop/diplom/sintez
$ ionic serve
Running live reload server: undefined
Watching: 0=www/**/*, 1=!www/lib/**/*
Running dev server: http://localhost:8100
Ionic server commands, enter:
  restart or r to restart the client app from the root
  goto or g and a url to have the app navigate to the given url
  consolelogs or c to enable/disable console log output
  serverlogs or s to enable/disable server log output
  quit or q to shutdown the server and exit

ionic $ |
```

Рис. 3 — Ввод команды ionic

Запускаем собранный проект. Для этого следует открыть папку синтез через консоль, и ввести команду: Ionic serve (рисунок 3).

В браузере открывается окно, имитирующее работу мобильного приложения. С помощью скроллинга стоит двигаться по странице и подгружать нужные элементы (рисунок 4).

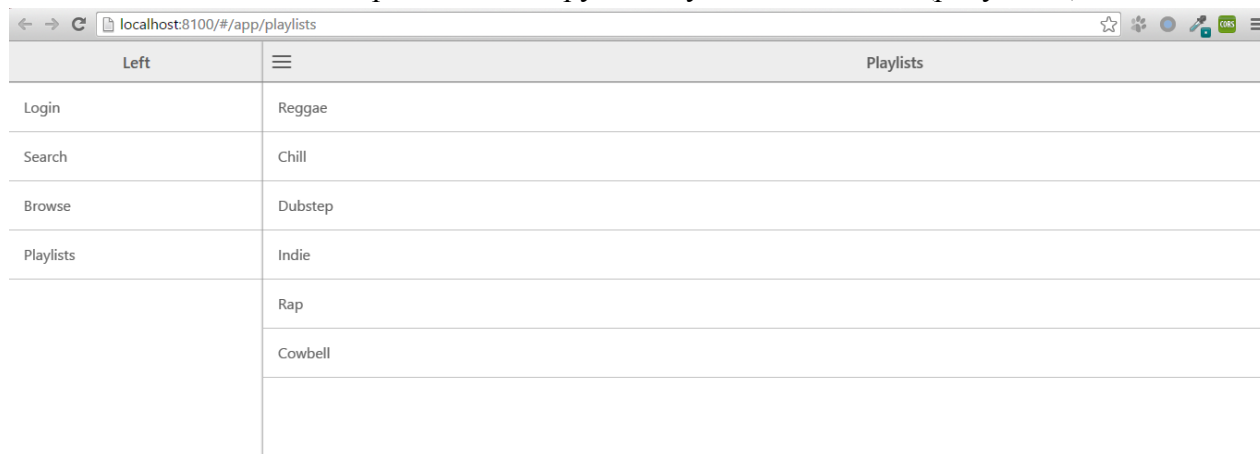


Рис. 4 – приложение с боковым меню

После выполнения всех вышеприведённых инструкций, можно приступить к разработке приложения. С помощью редактора программного кода ide WebStorm, добавляются необходимые html страницы, редактируется уже имеющийся код и изменяется заданный по умолчанию вариант в соответствии с поставленной задачей. Для создания и подключения страницы к приложению, достаточно зайти в «app.js» файл, который лежит в папке «js» и по подключённым элементам, как по шаблону, добавить имеющиеся наработки. В дальнейшем предыдущие подключённые файлы можно полностью удалить за ненадобностью. Это позволит уменьшить количество кода и файлов в проекте, что соответственно повлияет на вес файла. В дальнейшем эти файлы можно легко

восстановить. Для корректной работы приложения используются директивы angular.js (рисунок 5).

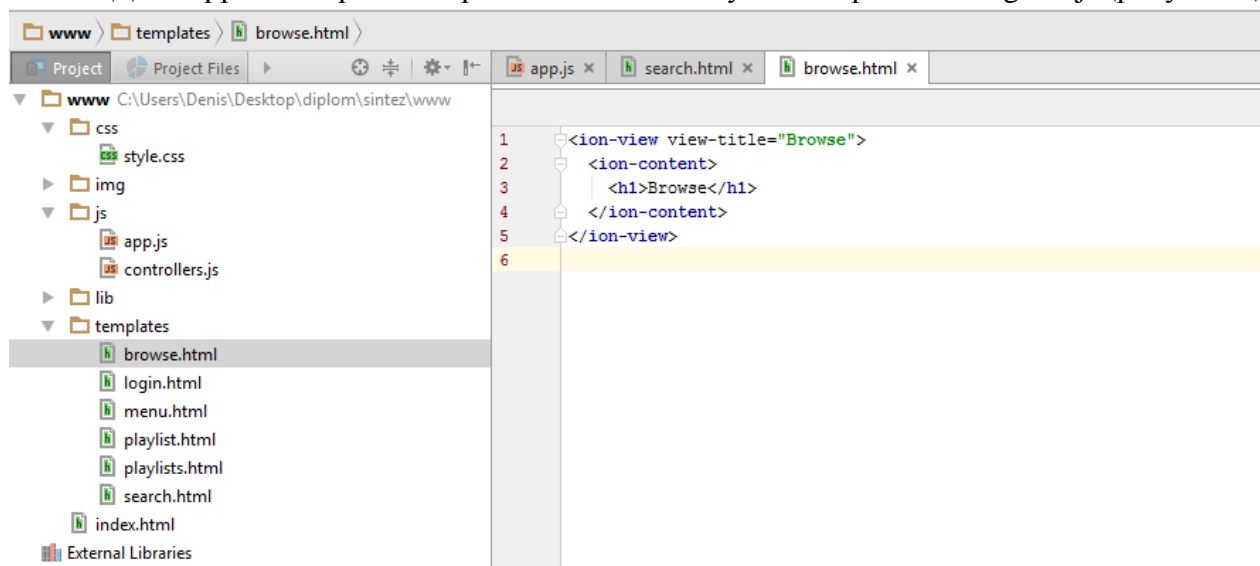


Рис. 5 — Содержимое html файлов

Во-первых, необходимо создать нужные страницы, которые будут использованы в приложении, а именно:

- About_company (информация о компании);
- All_AZS (все автозаправки);
- AZS_near (ближайшая автозаправка);
- Build_route (построить маршрут);
- Feedback (обратная связь);
- search_history (история поиска).

Создаём эти страницы в папке «templates» с расширением html. Теперь в каждом html документе нужно написать следующие директивы:

```
<ion-view view-title="О компании">
  <ion-content>
  </ion-content>
</ion-view>
```

где ion-view — навигационный заголовок бара, ion-content — содержимое страницы.

Для отображения файлов в приложении необходимо указать ссылки на документы в menu.html после объявления в app.js.

Index.html — одна из самых важным страниц в приложении. Она является связующим звеном всех директив и контроллеров. В index.html подключаем все нужные скрипты, ссылки на файлы, и API, необходимые для работы приложения.

Для получения доступа к Google картам [5], необходимо пройти регистрацию на официальном сайте и использовать полученный ключ (key) для API в создаваемом проекте. API — это набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах.

Получив API для сервера компании, пользователь может найти json-файл с информацией об автозаправочных станциях. Заполняем страницы контентом.

Далее приложение оформляется в цветовую гамму в соответствии с прототипом. С помощью средств разработчика, которые открываются в браузере клавишей F12, получается название класса объекта и прописываются в файле style.css нужные цвета — свойство background

[6]. После этого оформляется страница, исходя от прототипа. Получается следующий вид:

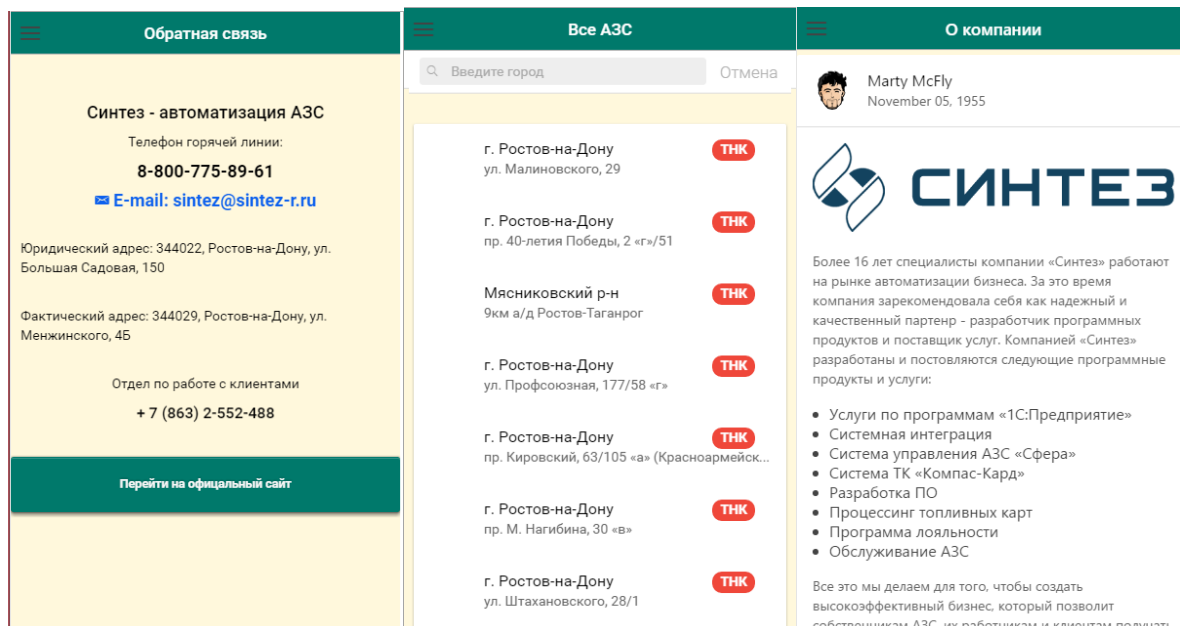


Рис.8 – Форма с информацией

Рис.к 9 – форма всех АЗС

Рис. 10 – форма обратной связи «Feedback»

о компании «About_company»

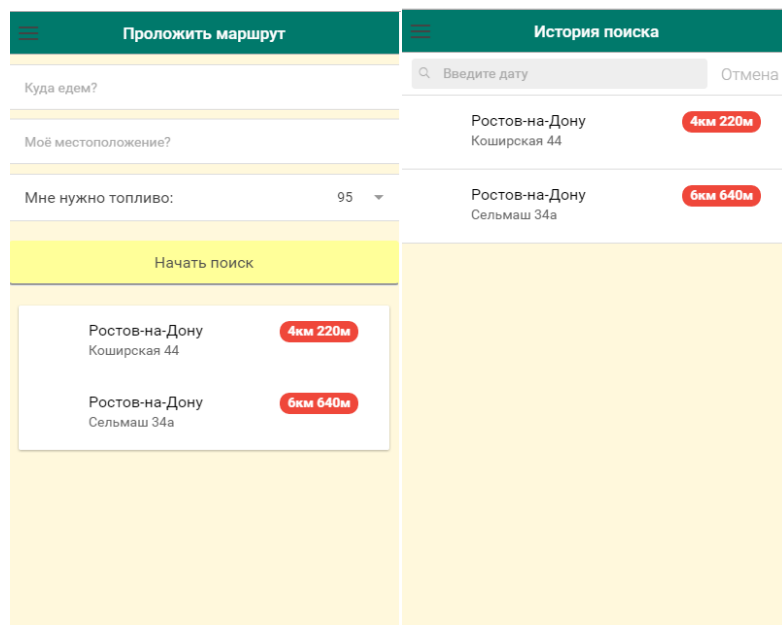


Рис. 11 – форма истории поиска «search_history»

Рис. 12 – форма постройки маршрута «Build_route»

В html-файле подготовится вёрстка для отображения данных. После проделанной работы можно увидеть большой список объектов (заправочных станций и информации по ним), где отображено самое важное: координаты в скрытом виде, id конкретной заправки в скрытом виде, название АЗС, город, адрес местонахождения, после обработки массива выводим данные в приложении. Чтобы упростить поиск по этому списку, а он состоит не менее чем из 450+ частей, следует создать поиск по объектам данной странички. Благодаря тому, что фреймворк Ionic содержит уже готовые компоненты, используется готовая вёрстка поиска с официального сайта:

<http://ionicframework.com/docs/> [7]. Далее необходимо найти нужный элемент, скопировать его исходный код и вставить в проект. По прописанным атрибутам классов `css`, вносятся поправки к элементам.

После того, как будет готов полученный таким образом код по странице, нужно уточнить его, чтобы поиск работал корректно. Когда вёрстка поиска будет успешно завершена необходимо скорректировать код под проект. В результате получится страница с результатом поиска необходимого объекта, то есть поиск АЗС по ключевым словам (рисунок 13) завершён.

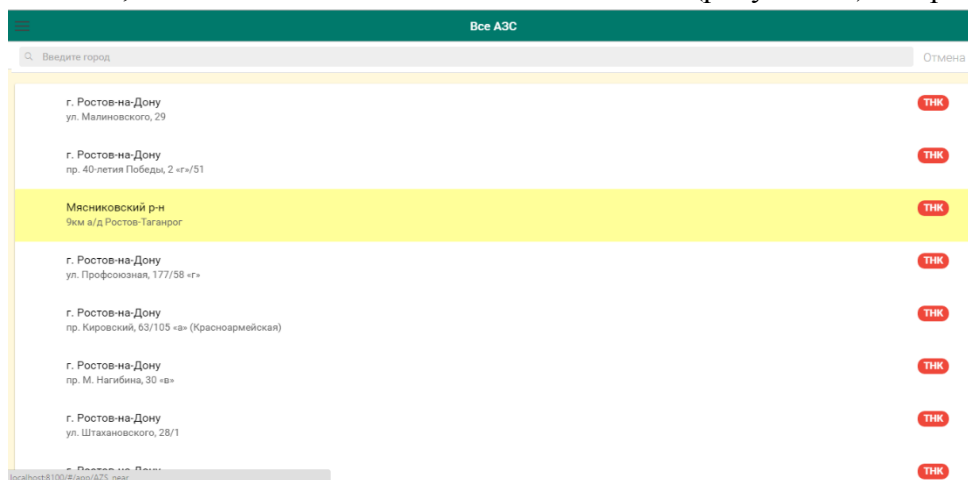


Рис. 13 — Все АЗС и поиск по ним.

Приступим к созданию карты в проекте. Для этого потребуется поиск по документации Google maps, с использованием `angular.js`, позволяющего решить данную задачу. Применяя найденный скрипт из официальной документации Google в `controllers.js`, на страницу `AZS_near.html` добавляется тег карты. С официального сайта Google копируем шаблон кода плагина, созданный для работы с картой и позволяющий строить маршрут от точки к точке. После действуем по инструкции официального сайта `Angular.js` [8]. Маркер размещается на местоположении водителя, второй маркер — на местоположении необходимого объекта. По обращению (клику) к выбранному объекту, по его `id` из `json`-файла берутся координаты и применяются ко второму маркеру. После этого строится маршрут от водительского маркера к заправочному маркеру (рисунок 15).

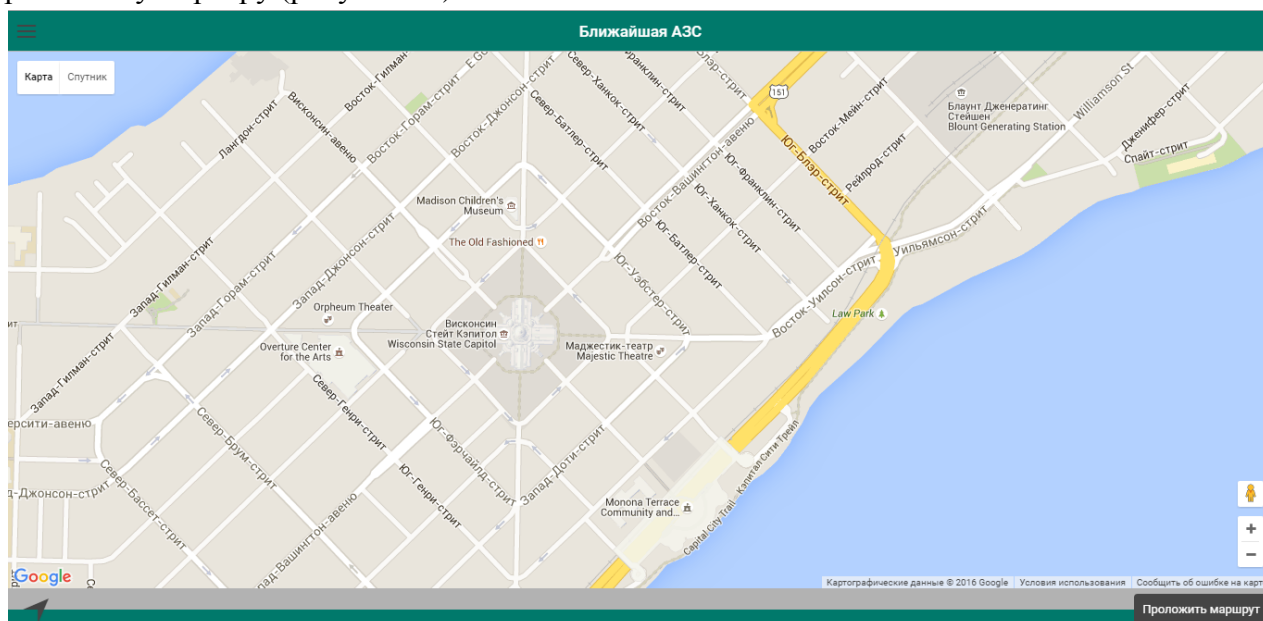


Рис. 14 – Подключение скрипта к контроллеру

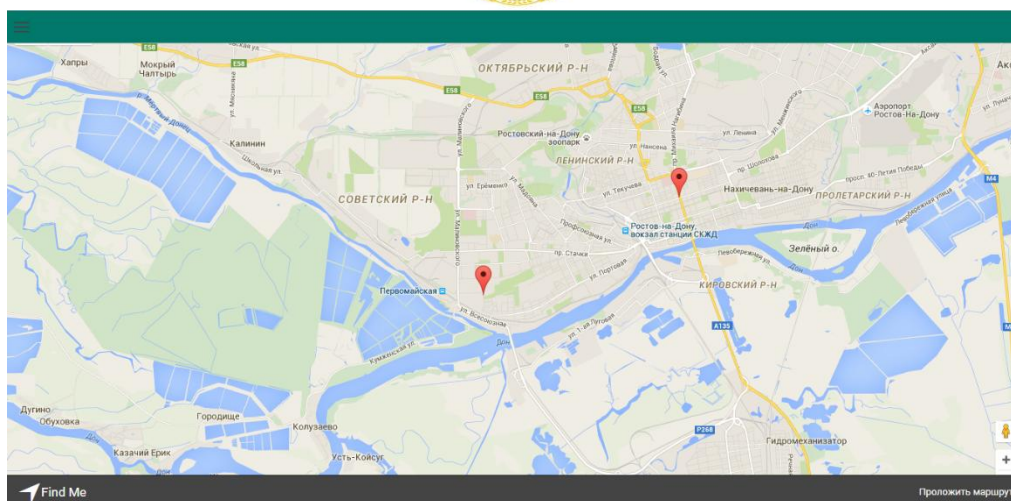


Рис.15 – Местоположение и обрисованная точка по координатам

Для поиска ближайшей АЗС используем следующую инструкцию: используется массив с точками; находится заданная точка; определяется расстояние между точками.

Готовый проект запускается в эмуляторе iOS и Android.

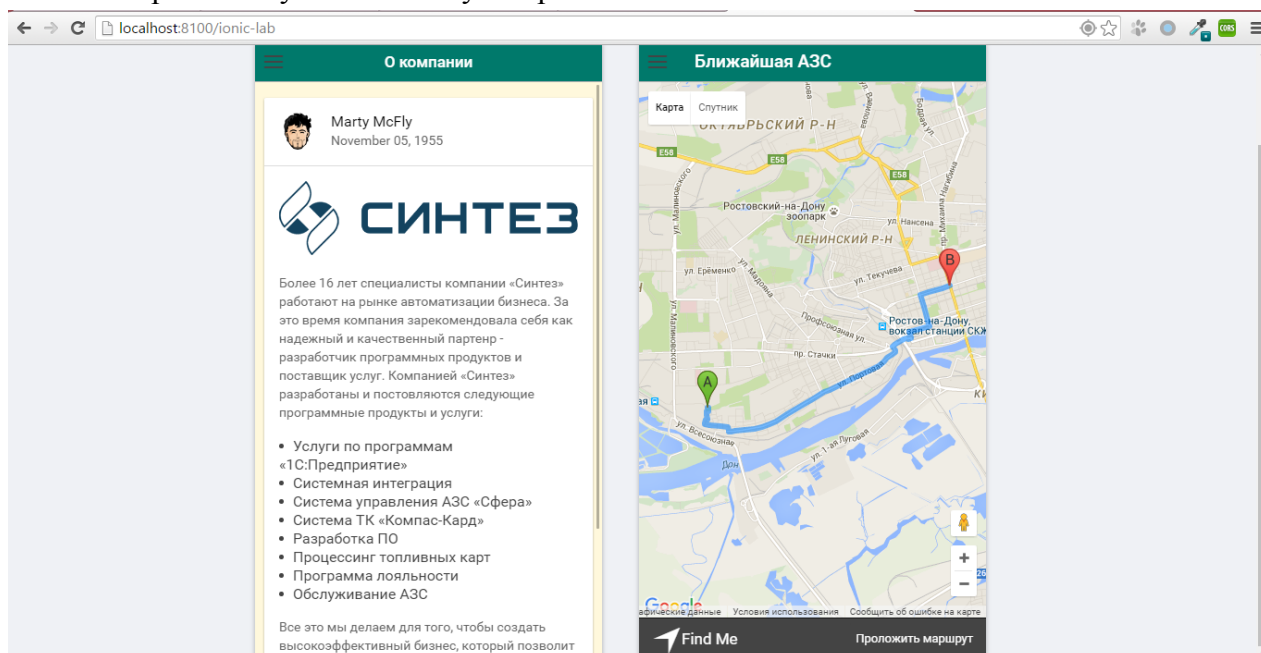


Рис. 16 – работа созданного приложения в среде ionic lab

Чтобы сконвертировать созданный проект, достаточно открыть папку проекта и в консоли прописать следующую команду: `ionic build android/ios`

Далее можно конвертировать приложение в установочный файл для Андроида и для iOS.

Получен файл, готовый к установке на смартфон или планшет. При подключении устройства к Интернету и геолокации, получается сервис поиска АЗС, что и являлось целью проекта.

Заключение. Основные результаты работы сформулированы в следующем виде:

- проведен анализ эффективности использования Интернет-технологий в создании мобильного приложений на фреймворке;
- итоговая информация представляется в виде программного продукта, отвечающего требованиям поставленной задачи;

- разработан удобный и интуитивно понятный для пользователя интерфейс;
- после создания, приложение было тщательно протестировано, на этапе завершения продукт является полностью рабочим и готовым к использованию.

Библиографический список.

1. Свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki> (дата обращения 06.11.2017).
2. ГОСТ ИЕС 60950-1. Оборудование информационных технологий. Требования безопасности. Общие требования. — Москва : Стандартформ, 2013. — 218 с.
3. Документация и информация о фреймворке angular.js [Электронный ресурс]. — Режим доступа: <https://angularjs.org/> (дата обращения 27.10.2017).
4. Документация и информация о фреймворке ionic [Электронный ресурс]. — Режим доступа: <http://ionicframework.com/> (дата обращения 15.10.2017).
5. Angular Google maps [Электронный ресурс] – Режим доступа: <http://angular-ui.github.io/angular-Google-maps/#!/> (дата обращения 06.11.2017)
6. Сайт для разработчиков, где можно найти ответ или задать свой вопрос [Электронный ресурс]. — Режим доступа: <http://stackoverflow.com/> (дата обращения 25.11.2017).
7. Сайт для разработчиков, где можно найти ответ или задать свой вопрос [Электронный ресурс]. — Режим доступа: <https://toster.ru/> (дата обращения 25.11.2017).
8. Maps javascript API [Электронный ресурс] — Режим доступа: <https://developers.Google.com/maps/documentation/javascript/> (дата обращения 27.10.2017).