

004.4

ОСНОВНЫЕ ПОДХОДЫ К РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЙ

Порошин И. А., Бужинская Н.В.

филиал РГППУ в г. Нижнем Тагиле, e-mail: Nadezhda_V_A@mail.ru

В настоящее время веб -приложения могут использоваться в различных сферах жизнедеятельности человека, например для организации взаимодействия пользователей, продажи товаров и услуг, организации взаимодействия с клиентами. Разработка веб-приложений является достаточно трудоемким процессом, который включает анализ предметной области, изучений предпочтений заказчика, определение требований к программному продукту, проектирование интерфейса и выбор среды разработки, внедрение и сопровождение продукта. Каждый разработчик должен уметь проанализировать возможности сред и языков программирования согласно выделенным критериям, поскольку от этого зависит продолжительность и результативность его работы. В данной статье рассматриваются основные подходы к разработке веб-приложений на примере планировщика маршрута по достопримечательностям в любом городе мира. Использование данного приложения позволит пользователю грамотно спланировать свой маршрут.

Ключевые слова: веб-приложение, проектирование, разработка, редактор кода.

MAIN APPROACHES TO DEVELOPING WEB APPLICATIONS

Poroshin IA, Buzhinskaya NV

Branch of the Russian State Pedagogical University in Nizhny Tagil,

e-mail: Nadezhda_V_A@mail.ru

Currently, web applications can be used in various spheres of human life, for example, for the organization of user interaction, the sale of goods and services, the organization of interaction with customers. Web application development is a fairly time-consuming process, which involves the analysis of the domain, the study of customer preferences, the definition of requirements for the software product, interface design and selection of the development environment, implementation and maintenance of the product. Each developer should be able to analyze the capabilities of the environments and programming languages according to the selected criteria, since this determines the duration and effectiveness of his work. This article examines the main approaches to the development of Web applications using the example of a route planner for sightseeing in any city in the world. Using this application will allow the user to plan their route correctly. Keywords: web application, design, development, code editor.

Keywords: web application, design, development, code editor.

На современном этапе развития информационно-коммуникационных технологий отмечается повышение требований к качеству веб-приложений. В отличие от веб-сайтов, которые используются, в основном, для предоставления информации, веб-приложения являются более интерактивными и позволяют пользователю принимать активное участие в работе [3]. Возможности современных веб-приложений сравнимы с возможностями

классических приложений, однако доступ к ним мы можем получить в любой момент времени при наличии доступа в Интернет.

Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети.

Работа над веб-приложением включает несколько этапов, которые соответствуют этапам жизненного цикла [1]:

- определение требований к приложению;
- проектирование приложения;
- разработка;
- тестирование проекта;
- внедрение проекта;
- сопровождение проекта.

Рассмотрим данные этапы более подробно.

Целью нашего проекта является разработка приложения, позволяющего планировать маршрут по достопримечательностям в любом городе мира. Данное приложение может использоваться как гидом при составлении экскурсионного маршрута по достопримечательностям для туристов, так и непосредственно самими туристами.

Для разработки данного приложения нам потребуется [3, 5]:

- интегрированная среда разработки или редактор-кода;
- язык гипертекстовой разметки HTML5;
- каскадные таблицы стилей CSS3;
- сценарный язык программирования JavaScript;
- набор библиотек и компонентов для упрощения разработки фреймворк.

При разработке выбираем «легкий» редактор-кода, а не интегрированную среду разработки [3]. Подавляющая часть ИСР – платные, потребляют много ресурсов ПК, чрезмерный функционал, для использования требуют достаточно времени в обучение.

Сравним редакторы кода, которые мы можем использовать для разработки приложения по следующим критериям (см. табл. 1) [6, 9, 10]:

- стоимость;
- наличие русифицированной версии;
- наличие обновлений;
- используемая платформа;

- скорость работы;
- функциональность.

Таблица 1

Сравнительная характеристика редакторов кода, которые могут использоваться для разработки приложения

Критерий/Название	Sublime Text	Atom	Visual Studio Code
Стоимость	Платный	Бесплатный текстовый редактор с открытым исходным кодом	Бесплатный кроссплатформенный редактор исходного кода
Русифицированная версия	-	-	+
Обновление	Нечасто	Активное обновление	Активное обновление
Платформы	Microsoft Windows, macOS и Linux	OSX, Windows, Linux, FreeBSD	Linux x86_64, OS X 10.9 или новее, Windows 7 или новее
Скорость работы (быстродействие)	Быстрый	Медленный Использует много памяти	Быстрый
Функциональность	Недостаточно, требуются дополнительные расширения	Недостаточно, требуются дополнительные расширения	Достаточно близко к IDE(интегрированная среда разработки)

Таким образом, для разработки нашего приложения был выбран редактор кода Visual Studio Code. Это кроссплатформенный редактор исходного кода, поддерживающий базовые возможности интегрированной среды разработки, созданный в Microsoft. Позиционируется как «легкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Visual Studio Code распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, и доступен в версиях для платформ Windows, Linux и OS X. В редакторе присутствуют встроенный отладчик, инструменты для работы с Git и средства рефакторинга, навигации по коду, автодополнения типовых конструкций и контекстной подсказки. Выбор редактора-кода, во основном индивидуален и зачастую зависит от проекта и разработчика в целом.

Кроме того, процесс разработки включает веб-дизайн, вёрстку страниц, программирование на стороне клиента и сервера, а также конфигурирование веб-сервера.

В приложение реализованы с помощью библиотек и вспомогательных инструментов следующие функции [4, 5, 7, 8]:

- работа с картой – Google Maps API Стандартный;
- подсказки мест Google Places API Web Service Стандартный;

- Drag-and-drop – jquery ui;
- маркер на карте – Google Places API Стандартный;
- построение маршрута – Google Directions API Стандартный.

```
1. // функция создания линии маршрута
2.  var polyline;
3.
4.  function showPath(map, locations) {
5.    if (polyline) {
6.      polyline.setMap(null);
7.    }
8.    var bounds = new google.maps.LatLngBounds();
9.    for (var i = 0; i < locations.length; i++) {
10.      bounds.extend(locations[i]);
11.    }
12.    map.fitBounds(bounds);
13.    polyline = new google.maps.Polyline({
14.      path: locations,
15.      strokeColor: '#FF0000',
16.      strokeOpacity: 1.0,
17.      strokeWeight: 2
18.    });
19.    polyline.setMap(map);
20.  }
21.
```

Рис. 1. Функция построения маршрута

Работу веб-приложения для планирования маршрута по достопримечательностям можно продемонстрировать на основе объектно-ориентированного подхода, языка UML. Моделирование с помощью UML предполагает построение диаграммы, которая позволяет продемонстрировать возможности приложения (см. рис. 1) [2].

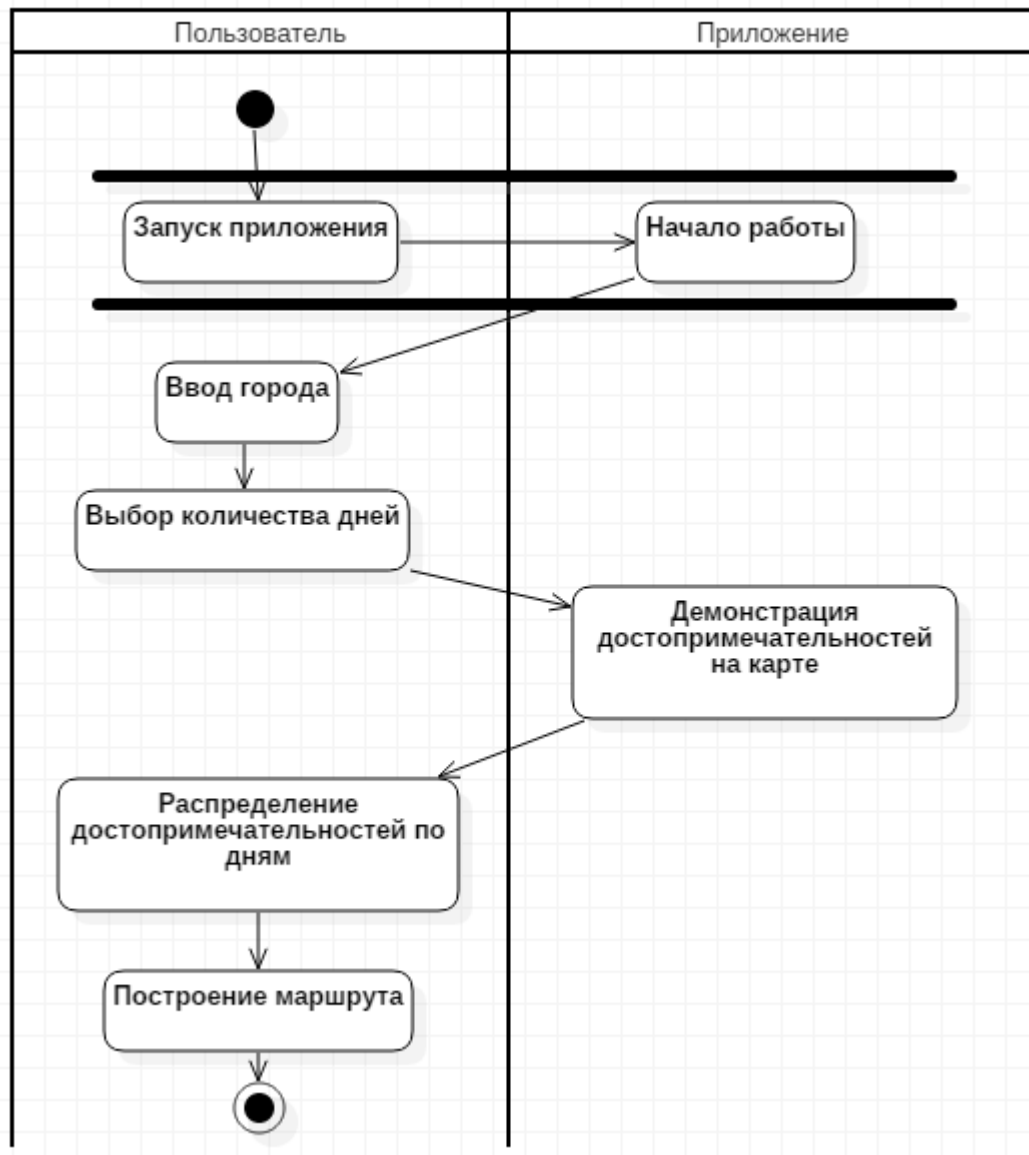


Рис. 2. Диаграмма деятельности, иллюстрирующая работу планировщика маршрута по достопримечательностям

Пользователь выбирает город и количество дней из предложенного списка. Далее приложение отображает достопримечательности данного города в списке и на карте, а также дни (см. рис 2). Пользователь добавляет («перетаскивает») достопримечательности из списка для каждого дня своей поездки. Далее он выбирает день и видит примерный маршрут (см. рис 3).

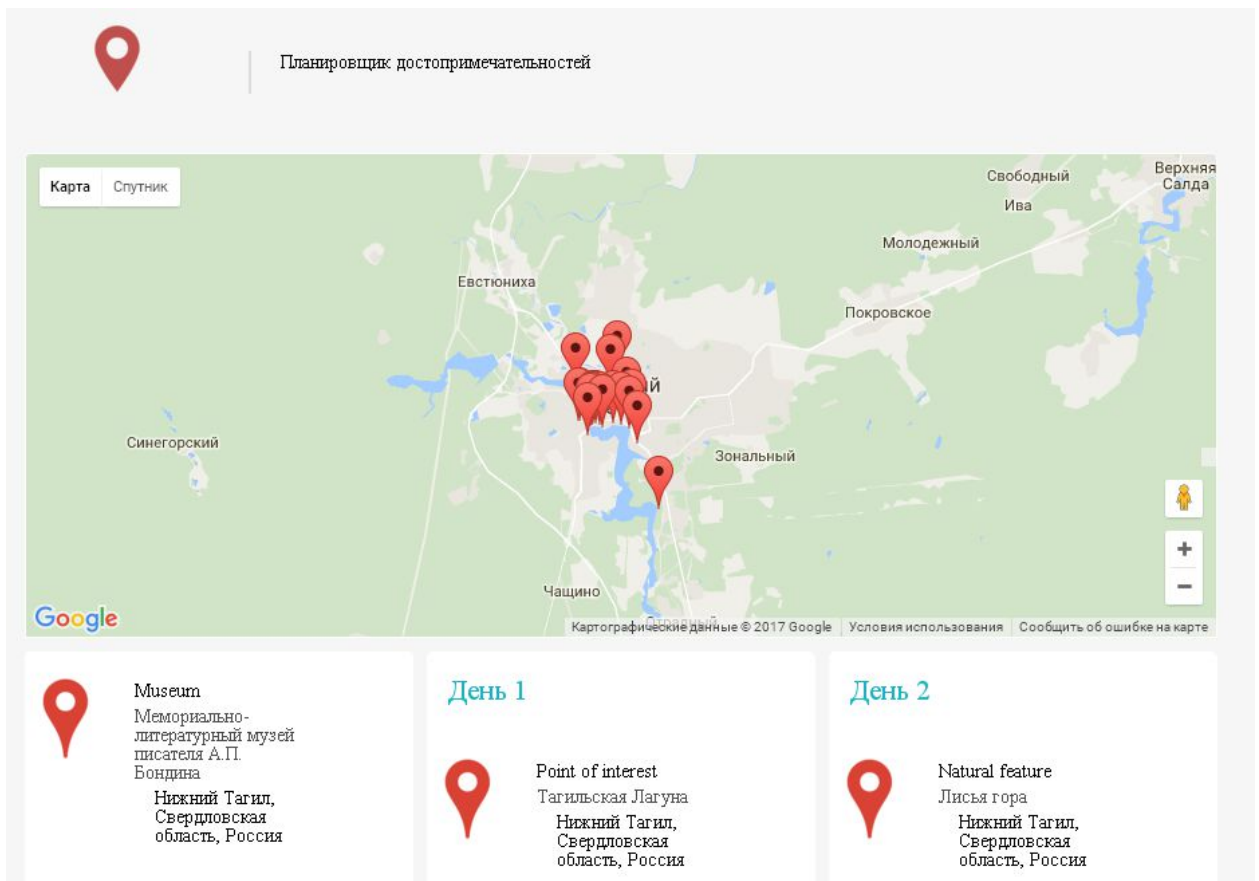


Рис. 3. Отображение достопримечательностей

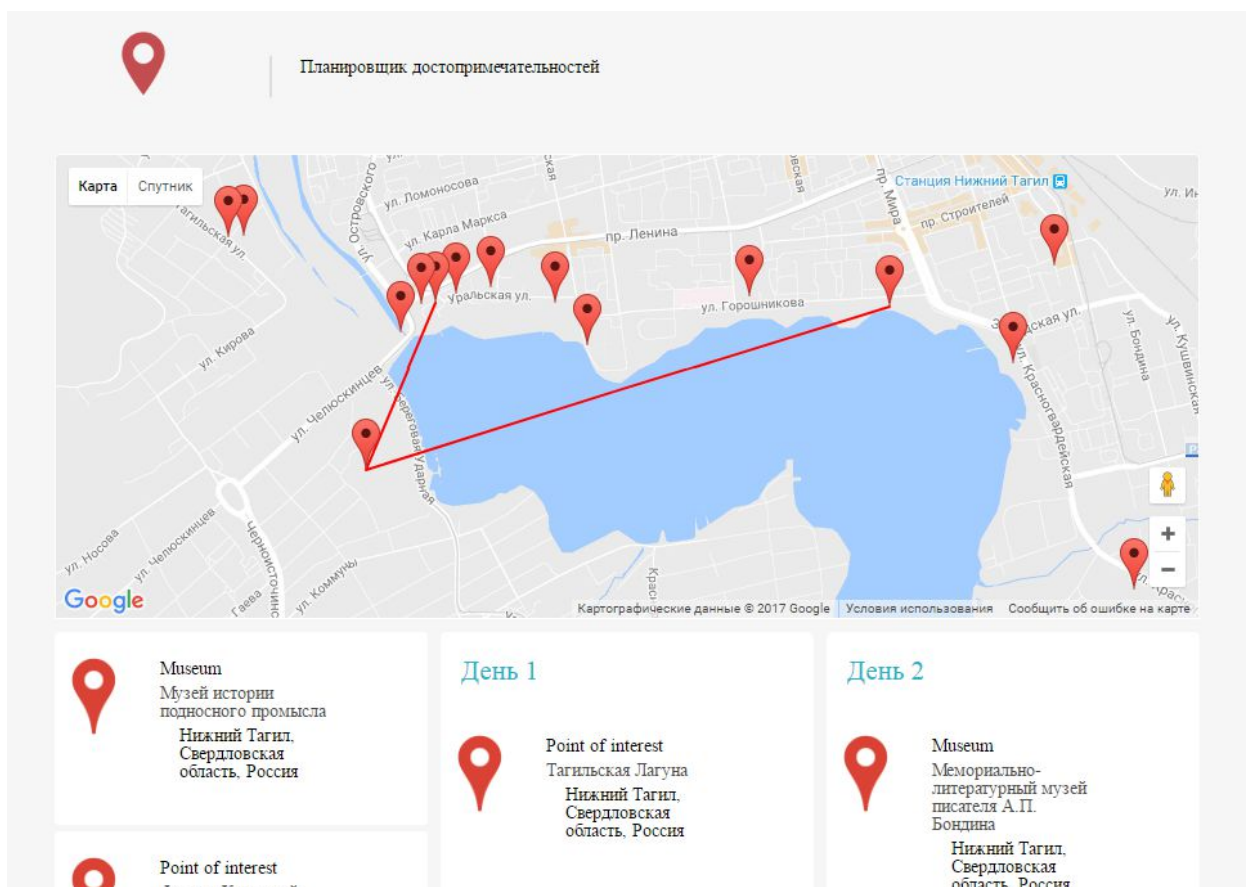


Рис. 4. Построение маршрутных линий

В процессе тестирования были выявлены недостатки, например, что маршрут можно построить как по прямой, так и по ломаным линиям с указанием длины расстояния. Кроме того, возможно реализовать сортировку по типу достопримечательности, рейтинг, стоимости и другим критериям. Данные требования могут быть рассмотрены, в дальнейшем, в качестве показателей для повышения функциональности приложения.

Список литературы

1. Автоматизированные системы. Стадии создания [Электронный ресурс]. URL: <https://www.franklin-grant.ru/ru/technologies/gost-34.601-90.shtml> (дата обращения: 24.05.2017).
2. Визуальное моделирование систем в StarUML: Учебное пособие. Казань. Казанский федеральный университет, 2013. 104с.
3. Никсон Робин. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Питер. 2016. 768 с.
4. Поиск Места | Google Places API [Электронный ресурс]. URL: <https://developers.google.com/places/web-service/search?hl=ru> (дата обращения: 13.04.2017).
5. Современный учебник JavaScript [Электронный ресурс]. URL: <https://learn.javascript.ru/> (дата обращения: 25.05.2017).
6. Atom [Электронный ресурс]. URL: <https://atom.io/> (дата обращения: 10.02.2017).
7. Google Maps API [Электронный ресурс]. URL: <https://developers.google.com/maps/?hl=ru> (дата обращения: 13.04.2017).
8. JQueryUI [Электронный ресурс]. URL: <https://jqueryui.com/> (дата обращения: 13.04.2017).
9. Sublime Text [Электронный ресурс]. URL: <https://www.sublimetext.com/> (дата обращения: 10.02.2017).
10. Visual Studio Code [Электронный ресурс]. URL: <https://code.visualstudio.com/> (дата обращения: 10.02.2017).