

УДК 004.457

## ПОДХОДЫ К ПРОЕКТИРОВАНИЮ ГРАФИЧЕСКОГО ВЕБ-ИНТЕРФЕЙСА НА ПРИМЕРЕ ВЕБ-ПРИЛОЖЕНИЯ «ВЕБ-ПРИЛОЖЕНИЕ ПО НЕКОММЕРЧЕСКИМ ПЕРЕВОЗКАМ ПассажиРОВ»

**Н.А. БОБКОВ***(Представлено: Д.Ф. ПАСТУХОВ)*

*Представлен анализ технологий, используемых для реализации интерфейсов веб-приложений. Приводится пример реализации веб-интерфейса одной из страниц веб-приложения. Показаны подходы к проектированию графического веб-интерфейса.*

С развитием Интернет-технологий стало появляться все больше веб-приложений, тематика которых совершенно различна.

Веб-приложение – клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по сети.

Веб-приложение состоит из клиентской и серверной частей, тем самым реализуя технологию «клиент – сервер». Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него [1].

Разработка веб-приложений – это мощный инструмент, который позволяет реализовывать бизнес-идеи и создавать успешные программные продукты. Именно веб-приложения помогают автоматизировать процесс работы с крупными потоками информации (клиентская база, каталог товаров, филиальная сеть, документы и т.д.).

Front-end и back-end – термины в программной инженерии, которые различают согласно принципу разделения ответственности между внешним представлением и внутренней реализацией соответственно. Front-end – это абстракция, которая предоставляет пользовательский интерфейс. Например, в проектировании программного обеспечения Model-View-Controller архитектура обеспечивает front-end и back-end между базой данных, компонентами обработки данных и пользователями [1].

Разработка веб-приложений по системе front – end и back – end подразумевает иерархическое разделение процесса создания ресурса на две части, на разработку пользовательского интерфейса (фронтэнда) и его программно-административной части (бэкэнда).

В настоящее время веб-приложения уже сопоставимы по своим возможностям с классическими приложениями. Но при этом могут быть доступны в любом месте и в любое время на компьютере, планшете или мобильном устройстве и зачастую имеют меньшую совокупную стоимость владения. Эти особенности делают веб-технологии очень привлекательными для решения широкого спектра задач.

**Средства решения задачи.** Рассмотрим основные средства реализации фронтэнда веб-приложения по некоммерческим перевозкам пассажиров.

Front – end разработка – это работа по созданию публичной части приложения, с которой непосредственно контактирует пользователь, и функционал, который обычно обыгрывается на клиентской стороне (в браузере). Составляющая часть приложения Front – end отвечает за вывод определенной информации пользователю и по факту совершения им каких-либо действий в веб-приложении, интерпретацию ее в вид, понятный программам, относящимся к бэкэнду.

В работе был использован Bootstrap – CSS/HTML фреймворк для создания веб-приложений, который содержит ряд преимуществ, благодаря которым он считается самым популярным среди себе подобных. Другими словами, это набор инструментов для верстки.

Рассмотрим основные преимущества Bootstrap:

- скорость работы – благодаря множеству готовых элементов верстка с бутстрапом занимает значительно меньше времени;
- масштабируемость – добавление новых элементов не нарушает общую структуру;
- легкая настраиваемость – редактирование стилей производится путем создания новых css-правил, которые исполняются вместо стандартных;
- большое количество шаблонов;
- огромное сообщество разработчиков;
- широкая сфера применения – Bootstrap используется в создании тем для практически любой CMS (OpenCart, Prestashop, Magento, Joomla, Bitrix, WordPress и любые другие), в том числе для одностраничных приложений [3]. Bootstrap использует самые современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

Основные инструменты Bootstrap:

- сетки – заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140 px относится к классу `.span2` (`.col-md-2` в третьей версии фреймворка), который можно использовать в CSS описании документа;
- шаблоны – фиксированный или резиновый шаблон документа;
- типографика – описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т.д.;
- медиа – представляет некоторое управление изображениями и видео;
- таблицы – средства оформления таблиц, вплоть до добавления функциональности сортировки;
- формы – классы для оформления форм и некоторых событий происходящих с ними;
- навигация – классы оформления для табов, вкладок, страничности, меню и тулбара;
- алерты – оформление диалоговых окон, подсказок и всплывающих окон [1].

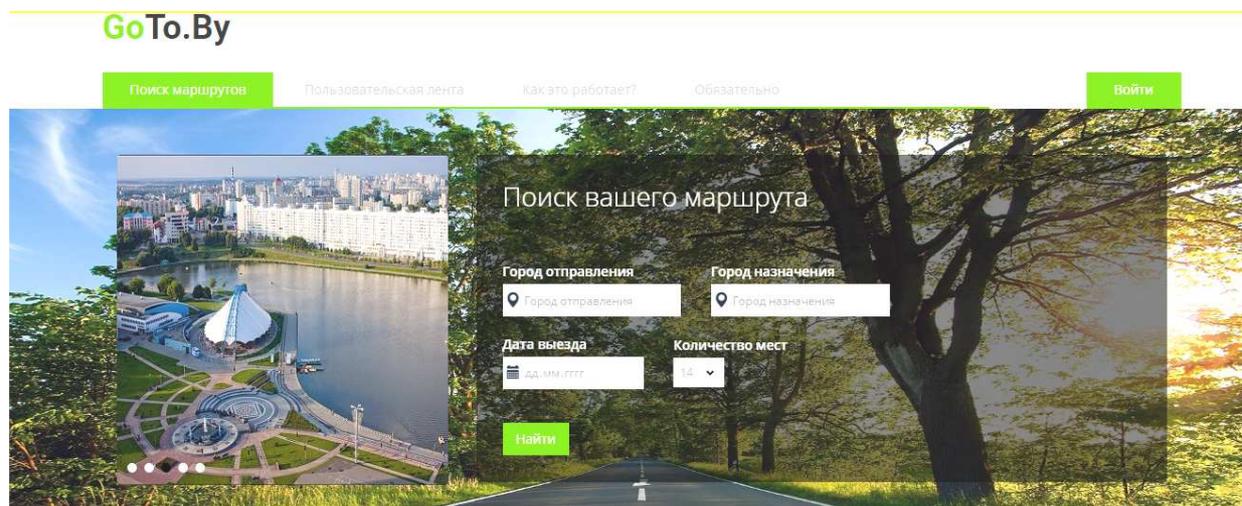
Также при реализации веб-приложения использовалась JSP (Java Server Pages) – технология, позволяющая создавать содержимое, которое имеет как статические, так и динамические компоненты. Страница JSP содержит текст двух типов: статические исходные данные, которые могут быть оформлены в одном из текстовых форматов HTML, SVG, WML или XML, и JSP-элементы, которые конструируют динамическое содержимое. Кроме этого могут использоваться библиотеки JSP-тегов, а также EL (Expression Language) для внедрения Java-кода в статичное содержимое JSP-страниц [1].

JSP является платформонезависимой, переносимой и легко расширяемой технологией для разработки веб-приложений. Основная идея JSP очень проста – сама страница представляет из себя шаблон с уже заготовленными HTML-тэгами, между которыми надо вставить нужные данные.

JSP страницы имеют расширение `.jsp` и размещаются там же, где и обычные web-страницы. Структура таких страниц может состоять из пяти конструкций: HTML, комментарии, скриптовые элементы, директивы и действия. JSP-страница при компиляции преобразуется в сервлет со статическим содержимым, которое направляется в поток вывода, связанный с методом `service`. Поэтому при первом запросе этот процесс может вызвать небольшую задержку. Скриптовые элементы позволяют указать код на языке Java, который впоследствии станет частью конечного сервлета, директивы дают возможность управлять всей структурой сервлета, а действия служат для задания существующих используемых компонентов, а также для контроля над поведением движка JSP. Для упрощения работы со скриптами имеются заранее определенные переменные, такие как `request`, `response`, `pageContext`, `session`, `out`, `application`, `config`, `page`, `exception` [1; 2].

Код JSP-страницы транслируется в Java-код сервлета с помощью компилятора JSP-страниц Jasper, а затем компилируется в байт-код виртуальной машины java (JVM). Контейнеры сервлетов, способные исполнять JSP-страницы, написаны на языке Java. JSP-страницы загружаются на сервере и управляются из структуры специального Java server packet, который называется Java EE Web Application. Обычно страницы упакованы в файловые архивы `.war` и `.ear`.

**Проектирование интерфейса.** В главном окне приложения можно осуществить поиск маршрутов без просмотра контактной информации водителей, просмотреть последние двадцать фотографий, загруженные пользователями, а также изучить принципы работы системы (рисунок).



Пожалуйста, пройдите аутентификацию! [Сюда!](#)

Главное окно приложения

При входе на сайт приложения для каждого пользователя создается уникальная сессия, которая сохраняется на всем протяжении нахождения и пользования данным ресурсом. До того момента, пока пользователь не авторизовался, ему будет присвоена анонимная роль, в соответствии с которой он может просматривать только закладки, расположенные на главной странице. После авторизации пользователю добавятся те или иные полномочия, в зависимости от его роли в системе.

При выборе пункта меню «Пользовательская лента» перед пользователем появляется окно, которое содержит последние двадцать добавленных фотографий с описанием местоположения съемки.

На вкладке «Как это работает» пользователь может узнать о работе системы в целом. Данная вкладка содержит следующие пункты:

- для водителей – описание действий водителя, для комфортной поездки;
- для попутчиков – описание действий попутчиков, для удобства при переезде.

Вкладка «Обязательно» несет информационную задачу. Её цель – донести до пользователей, что будет, если нарушить политику приложения.

При выборе пункта меню «Войти» перед пользователем появляется окно, которое можно использовать как при авторизации в веб-приложении, так и для регистрации нового аккаунта. При нажатии на надпись «Забыли пароль» приложение переведет ваш запрос на страничку, содержащую всего одно текстовое поле для ввода адреса электронной почты, к которой привязан пользовательский аккаунт.

Если у пользователя не существует аккаунта в системе, он должен будет нажать на надпись: «Зарегистрируйтесь». В появившемся окне регистрации, в левой части экрана, расположены две кнопки для быстрой регистрации через социальные сети «Вконтакте» и «Facebook». В правой части экрана расположена стандартная регистрация в приложении. При регистрации проверяется оригинальность псевдонима, используемого на сайте при последующей авторизации, и адрес электронной почты.

По окончании любой регистрации перед пользователем появится окно, информирующее его об успешном создании аккаунта в приложении.

После успешной регистрации или авторизации в приложении пользователь перенаправляется в личный кабинет для редактирования информации о себе. Здесь в левой части экрана расположена фотография пользователя. При использовании быстрой регистрации фотография заимствуется из социальной сети, а при использовании стандартной регистрации веб-приложение устанавливает фотографию по умолчанию. Пользователь может изменить ее в любое удобное время. В правой части экрана располагается блок личной информации. При первом входе необходимо заполнить адрес электронной почты и контактный номер телефона для получения доступа к поиску маршрутов. Также при использовании быстрой регистрации рекомендуется изменить пароль, так как система устанавливает пароль по умолчанию «goto.by».

Пользователь может отметить свои предпочтения в поездке. При поиске маршрута по предпочтениям они будут учитываться.

В заключение отметим, что в настоящее время веб-технологии быстро развиваются, они дают разработчикам все больше возможностей, а их производительность растет. В итоге уменьшаются преимущества нативных приложений, а разработка все больше уходит в веб-среду. Веб-приложения прочно вошли в нашу жизнь, и многими из них мы пользуемся, даже не задумываясь, например поиском Google или почтовым сервисом Gmail.

При разработке веб-интерфейсов необходимо брать во внимание не только хороший дизайн, который помогает сконцентрироваться на задачах пользователя, но и учитывать то, что любой интерфейс должен быть интуитивно понятным для пользователя – он должен моментально понимать, какой следующий шаг он может совершить. Веб-интерфейс должен обладать такими качествами, как кроссплатформенность, адаптивность для различных размеров экранов, динамичность и корректная работа в различных браузерах.

Очевидно, что плюсов от использования веб-приложений гораздо больше, чем минусов, а использование различных фреймворков и технологий позволяет значительно сократить время реализации веб-приложений без потери их качества.

#### ЛИТЕРАТУРА

1. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/>. – Дата доступа: 23.09.2016.
2. Metanit – сайт о программировании [Электронный ресурс]. – Режим доступа: <http://metanit.com>. – Дата доступа: 23.09.2016.
3. HineX – компания, специализирующаяся в веб-разработке и Интернет-рекламе [Электронный ресурс]. – Режим доступа: <http://hinex.ru/>. – Дата доступа: 23.09.2016.