

УДК 004.457

## ПОДХОДЫ К ПРОЕКТИРОВАНИЮ ГРАФИЧЕСКОГО ВЕБ-ИНТЕРФЕЙСА НА ПРИМЕРЕ АНДРОИД-ПРИЛОЖЕНИЯ ДЛЯ ВЗАИМОДЕЙСТВИЯ С КОМПЬЮТЕРОМ ПО ПРОТОКОЛУ HTTP

*А.А. ЮШКЕВИЧ**(Представлено: Е.Р. СУХАРЕВ)*

*Анализируются технологии, которые часто используются при проектировании веб-интерфейсов. Приведен пример построения веб-интерфейса андроид-приложения, с помощью которого можно выполнять некоторые действия на мобильном телефоне.*

Развитие информационных технологий принесло в наш мир огромное количество электронных устройств. Это компьютеры, ноутбуки, мобильные телефоны, планшеты и т.д. Поэтому при разработке приложений важным является вопрос кроссплатформенности. Часто он решается разработкой одного и того же приложения под разные архитектуры с помощью компиляции в байт-код (например, в языке программирования Java). Однако стоит отметить, что сейчас практически для каждого устройства и операционной системы существует веб-браузер. И самое главное, им пользуются практически все. Поэтому приложение с веб-интерфейсом может быть запущено на большинстве устройств и его можно считать кроссплатформенным.

**Средства решения задач.** Рассмотрим подходы к проектированию интерфейса на примере разработанного андроид-приложения, которое позволяет установить взаимодействие между мобильным телефоном и компьютером.

Разработка андроид-приложения для взаимодействия с компьютером по протоколу HTTP является актуальной в связи с высокой популярностью мобильных устройств. А большую долю рынка занимает операционная система Андроид.

Взаимодействие между устройствами стоит рассматривать с точки зрения обмена файлами, просмотра и редактирования сообщений и контактов и прочих возможностей. Многие существующие в настоящее время решения могут оказаться платформенно-зависимыми. Даже с использованием кабеля решение задачи может оказаться проблемным при попытке взаимодействия с Linux-дистрибутивом по причине того, что драйвера, необходимые для какого-либо взаимодействия, за исключением обмена файлами, могут оказаться только для операционной системы Windows.

Разработка клиентской части с помощью веб-интерфейса может сделать этот элемент платформенно независимым. То есть для работы с приложением понадобится всего лишь соединение, например, с помощью Wi-Fi, а также веб-браузеры, которых сейчас огромное количество, и они разработаны практически для каждой операционной системы.

Современные технологии позволяют легко сделать интерфейс дружелюбным. При его разработке было решено использовать jQuery, Bootstrap и ajax.

При разработке интерфейса было решено, что большая часть информации будет загружаться не при загрузке страницы, а по мере необходимости. Это необходимо для уменьшения объема передаваемых данных.

Для облегчения работы с javascript была применена библиотека jQuery [1]. Главное её преимущество состоит в том, что разрабатывать с её помощью очень просто, так как можно использовать селекторы для выбора интересующих элементов, события. Также код получается намного короче, что позволяет повысить его читабельность, и в случае возникновения ошибок они могут быть быстрее обнаружены. А самое главное то, что разработчик может разрабатывать не задумываясь, под каким браузером код будет исполняться. Ведь кто-то использует Microsoft Edge, а кто-то браузеры на WebKit, такие как Opera, Google Chrome и т.д. У каждого из них есть свои особенности, которые должен учитывать разработчик, разрабатывая на «голом» javascript [2].

Bootstrap – это фреймворк, который разработала Twitter Inc. Он предназначен для облегчения построения графического интерфейса. Сама библиотека включает в себя огромное количество элементов, таких как кнопки, веб-формы, блоки навигации, и многое другое. Конечно, если изначально ничего не менять, то полученная форма будет похожа на то, что получается у многих других разработчиков, использующих аналогичный фреймворк, но следует учитывать, что на ранних этапах разработки может быть

получен достаточно аккуратный интерфейс, и в итоге его всё равно можно настроить. Исходные коды распространяются и лицензией MIT, что позволяет свободно использовать и изменять технологию [3].

Технология ajax предназначена для выполнения запросов на сервере с целью получения динамического контента, не перезагружая всю страницу. Это приводит к тому, что запросы выполняются быстрее, а соответственно, улучшается реакция на действия пользователя. В настоящее время эта технология применяется практически везде. Многие популярные приложения используют XMLHttpRequest [4]. Очень часто в качестве примера приводят сервисы Google. Например, Google Maps получает необходимую информацию налету, едва пользователь куда-либо нажмёт [5].

**Проектирование интерфейса.** При проектировании интерфейса решено сформировать несколько страниц, на каждой из которых реализуется какой-то конкретный функционал, а весь контент генерируется динамически посредством ajax-запросов.

Таким образом, сформированы страницы для работы с файлами, контактами, сообщениями, приложениями и фотографиями. При необходимости вывода какой-либо вспомогательной информации использовались модальные окна. Код самого окна заранее помещается на веб-страницу, но оно скрыто. При необходимости просмотра какой-либо информации, например списка сообщений какого-либо пользователя или подробной информации о контакте, выполняется ajax-запрос на сервер, где можно получить необходимую информацию. Затем она помещается в поле модального окна, после чего окно становится видимым [6].

Следует отметить, что в подобный элемент можно помещать progress bar для отображения состояния загрузки файлов на сервер.

Чтобы пользователь мог сразу видеть все возможности приложения, все кнопки действий расположили в левом столбце, а переключение между вкладками поместили в верхней части. Во вкладках было решено разделять глобальные действия, такие как работа с файлами, работа с сообщениями, работа с контактами, а также возможность работать с приложениями и фотографиями. Причём за каждую вкладку отвечает своя страница.

На рисунке 1 представлен пример формы для работы с файлами.

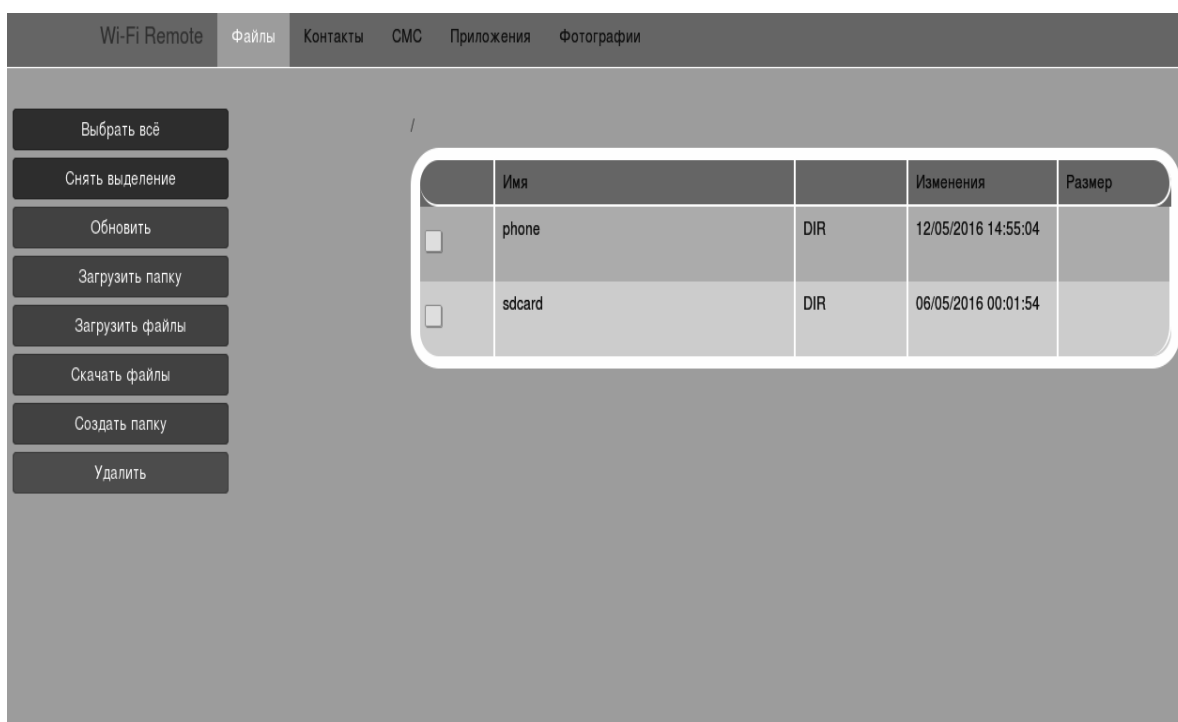


Рисунок 1. – Разработанный интерфейс на примере формы для работы с файлами

Как было сказано выше, располагаются вкладки для переключения между основными функционалами, а слева – кнопки для выполнения действий над текущей страницей (выбрать всё, снять выделение, обновить) или над файлами (загрузка файла, загрузка папки, скачивание папки, скачивание файла, создание папки, удаление файла или папки).

В основной части вверху находится строка, указывающая, в каком каталоге сейчас находится пользователь. Ниже изображена таблица, в которой содержится перечень файлов и папок в текущем каталоге, а также информация о них (тип, дата последнего изменения, размер файла). При загрузке файлов на сервер появляется модальное окно, отображающее прогресс.

При щелчке по строке, выполняется переход в папку, либо скачивание файла. При щелчке по чексбок-у, выполняется выделение или снятие выделения элемента. Выделенные элементы подсвечиваются.

Для построения панели с именами вкладок, функциональных кнопок в левой части, таблицы и для реализации эффекта чередующегося цвета строк в ней применялся фреймворк Bootstrap. Для подсвечивания отмеченных строк были использованы функции jQuery.

Для перехода по папкам информация собиралась с сервера без перезагрузки страницы с использованием ajax. Также с его помощью вызывается выполнение функций, возложенных на кнопки.

Остальные страницы разработаны в аналогичном стиле.

**Заключение.** Проведенное исследование позволяет сделать вывод, что правильным подходом при разработке веб-интерфейса является использование фреймворков, которые позволяют выполнять свои задачи во всех браузерах. Это значительно экономит время при разработке интерфейса, которое можно потратить на реализацию функциональной части.

Предложен процесс построения интерфейса и динамическая обработка запросов без перезагрузки страницы с помощью фреймворков jQuery, Bootstrap и технологии ajax.

#### ЛИТЕРАТУРА

1. jQuery / The jQuery Foundation [Electronic resource]. – 2016. – Mode of access: <https://jquery.com>. – Date of access: 20.09.2016.
2. W3Schools Online Web Tutorials / Refsnes [Electronic resource]. – 1999–2016. – Mode of access: <http://www.w3schools.com>. – Date of access: 20.09.2016.
3. Bootstrap / Twitter Inc. [Electronic resource]. – 2016. – Mode of access: <http://www.w3schools.com>. – Date of access: 20.09.2016.
4. XMLHttpRequest Level 1 / W3C [Electronic resource]. – 2014. – Mode of access: <http://www.w3schools.com>. – Date of access: 20.09.2016.
5. AJAX – Викиучебник / Wikimedia Foundation Inc. [Электронный ресурс]. – 2016. – Режим доступа: <https://ru.wikibooks.org/wiki/AJAX>. – Дата доступа: 20.09.2016.
6. Модальное окно Bootstrap Modal / htmlhook.ru [Электронный ресурс]. – 2016. – Режим доступа: <http://htmlhook.ru/modal-popup-bootstrap-3.html>. – Дата доступа: 20.09.2016.