

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ HTML5 ДЛЯ ЗАДАЧ ОБМЕНА МЕДИЦИНСКИМИ ДАННЫМИ

Бабаков И.В., Пономарев А.А.

Томский политехнический университет, 634050, Россия, Томск, ул. Ленина 30

InstanT.977@gmail.com

ziganshin@sibmail.com

Введение

В настоящее время в России, как и во всем мире, идет бурное внедрение информационных технологий во все сферы человеческой деятельности, в том числе и в медицинской отрасли. Хорошо известно, что в силу своих профессиональных обязанностей медицинским работникам требуется посещать пациентов и вести свою профессиональную деятельность за пределами ЛПУ, но, как правило, МИС не способны полноценно функционировать при отсутствии подключения к сети (*offline*).

В данный момент на рынке имеются программные продукты, позволяющие решить имеющуюся проблему. Но у каждой из них есть свои отрицательные стороны - не один из имеющихся на рынке продуктов не удовлетворяет в полной мере исходным данным.

Ни один из имеющихся на рынке продуктов не позволяет выполнять работу в режиме *offline* с МИС на планшетных устройствах, персональных компьютерах, ноутбуках одновременно. Одной из современных технологий, позволяющих реализовать такую задачу является HTML5.

От теории к практике - Web Sql

В HTML5 есть много новых возможностей, которые позволяют web разработчикам создавать мощные и насыщенные приложения. Для решения нашей задачи наибольший интерес представляют способы хранения данных на клиенте в *offline* режиме. К этим таким возможностям относятся Web storage и Web SQL database [1].

Первым шагом работы с Web SQL является инициализация пространства имен в html5.

```
var html5rocks = {};  
html5rocks.webdb = {};
```

База данных *html5* поддерживает как асинхронный, так и транзакционный режимы работы [4]. В транзакционном режиме также имеется два типа транзакций:

- Чтение/Запись (*transaction()*), при обращении к базе данных, не происходит блокирования записей;
- Только чтение (*readTransaction()*), при обращении к базе данных, происходит блокирование записей для запрета внесения изменений в данные.[1]

Чтобы получить доступ к базе данных необходимо открыть соединение, то есть указать имя

сервера, версию, а также текстовое описание и предполагаемый размер базы данных в битах.

```
html5rocks.webdb.db = null; // Инициализируем  
базу  
html5rocks.webdb.open = function() { // Открываем  
соединение  
var dbSize = 5 * 1024 * 1024; // 5MB  
html5rocks.webdb.db = openDatabase("Todo", "1",  
"Todo manager", dbSize);}  
html5rocks.webdb.onError = function(tx, e) {  
alert("Произошла ошибка: " + e.message);} // обра-  
ботчик ошибки onError  
html5rocks.webdb.onSuccess = function(tx, r) {  
html5rocks.webdb.getAllTodoItems(loadTodoItems);}  
// обработчик ошибки onSuccess
```

Теперь перейдем к созданию таблицы. Создать таблицу можно с помощью выполнения SQL команды *create table* [2]. Определим функцию, которая создаст таблицу с именем *todo* в случае её отсутствия в базе данных. Пусть таблица будет иметь три столбца:

- *ID* – инкрементное, уникальное поле;
- *todo* – текстовое поле;
- *added_on* – поле, содержащее время добавления записи в базу данных.

```
html5rocks.webdb.createTable = function() {  
var db = html5rocks.webdb.db; //Инициализируем  
объект базы данных  
db.transaction(function(tx) { // транзакционный за-  
прос на создание таблицы  
tx.executeSql("CREATE TABLE IF NOT EXISTS " +  
"todo(ID INTEGER PRIMARY KEY ASC, todo TEXT,  
added_on DATETIME)", [], {});}
```

Теперь, когда у нас имеется созданная таблица, мы можем наполнить её данными, используя SQL команду *insert*. [1]

Также в Web SQL существует команда *executeSql*, которая позволяет передавать в запросы в качестве параметров значения переменных, аналогично *String.Format* в C#, а также функции, которые выполняться в случае успешного запроса (*OnSuccess*), либо в случае ошибки (*onError*). Для добавления в базу данных записей из пользовательского интерфейса можно воспользоваться функцией «*html5rocks.webdb.addTodo*»

```
function addTodo() {  
var todo = document.getElementById("todo");  
html5rocks.webdb.addTodo(todo.value);  
todo.value = "";}[2]
```

Создание наилучшей обертки приложения

Конечно, самый простой вариант – использование браузера в качестве инструмента для работы с разрабатываемым приложением. Но тогда возникает проблема «кроссбраузерности», ведь одни браузеры поддерживают определенный набор html5 функций, а другие совершенно другой. Безусловно, спустя какое-то время эта проблема исчезнет сама по себе, в связи с развитием как самого html5, так и браузеров. Поэтому, чтобы получить полноценное приложение используя данные ресурсы, потребуется использование какого-либо SDK. Существует огромное количество разнообразных SDK позволяющих разрабатывать приложения на IOs и Android без знания Objective-C и Java языков программирования. Наиболее популярными считаются Marmelade, Xamarine, PhoneGap (Cordova). Остановим внимание на PhoneGap SDK подходящей по всем параметрам для решения поставленной задачи. [3] [4]

Это OpenSource платформа, позволяющая разрабатывать мобильные приложения на HTML5, JavaScript и CSS под различные платформы (практически без изменения кода приложения) в их число входят: iOS, Android, Blackberry, WebOS, Symbian и Windows Mobile. Основным его достоинством является то, что он не требует навыков разработки под конкретную платформу. Вы пишете свое приложение на JavaScript, используете HTML5 и CSS для разметки. Вы пишете мобильное приложение как обычный сайт или веб-сервис. Движок PhoneGap расширяет API браузера и добавляет следующие возможности: доступ к акселерометру,

- доступ к камере;
- доступ к компасу;
- доступ к списку контактов;
- запись и прослушивание аудио файлов.

Также предоставляет доступ к файловой системе, позволяет работать с разными HTML5 хранилищами localStorage, Web SQL и т.п. [5]

Используя данный SDK мы имеем доступ к его исходным кодам, а также очень гибкий дизайн, мгновенно изменяемый используя CSS и

абсолютно не привязанный к типу устройства. А также заметно упростит использование средств HTML5 для кеширования и работы с базой данных. При этом данная SDK развивается.

Заключение

Таким образом, HTML5 обладает такими технологиями как: *Application Cache*, *WebStorage* и *WebSql*, которые в совокупности позволяют в полной мере решить поставленную задачу. На примере, стало ясно, что база данных, поддерживаемая HTML5 – «Web SQL» является полноценной базой данных, с возможностью транзакций, а также асинхронным режимом работы. А для того чтобы сформировать полноценное нативное приложение, принято решение использовать PhoneGap SDK, позволяющий обернуть имеющиеся возможности связки JavaScript и HTML5 в один контейнер с очень масштабным функционалом и очень гибким интерфейсом.

Литература

1. Хабрахабр // HTML5. Работа с WebSql базой [Электронный ресурс]. - Режим доступа: <http://habrahabr.ru/post/84654/>, свободный.
2. Хабрахабр // «Перезагружаем в офлайн: WebStorage, Application Cache и WebSql [Электронный ресурс]. - Режим доступа: <http://habrahabr.ru/post/117123/>, свободный.
3. Машнин, Eclipse: разработка RCP-, Web-, Ajax- и Android - приложений на Java : . — Санкт-Петербург: БХВ-Петербург, 2013. — 380 с.: ил. + . — . — Библиогр.: с. — ISBN 978-5-9775-0829-2.
4. Marmelade SDK // Marmelade SDK – cross-platform app development tool [Электронный ресурс]. - Режим доступа: <https://www.madewithmarmalade.com>, свободный.
5. Applifto // Интересные новости мира Apple, приложений для iPhone и iPad [Электронный ресурс]. - Режим доступа: http://applifto.ru/article/podrobno_o_xamarin, свободный.
6. Argo // Argo. Введение в разработку [Электронный ресурс]. - Режим доступа: <http://www.coolmobmasters.com/phonegap-sdk/293-phonegap-development.html>, свободный.