



**«Московский государственный технический университет
имени Н.Э. Баумана (национальный исследовательский
институт)»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

О т ч е т п о п р о и з в о д с т в е н н о й п р а к т и к е

Студент: Куклина Нина Ильинична

Группа: ИУ6(61)

Название практики: технологический практикум

Название предприятия: Samsung

Руководитель от МГТУ им. Н. Э. Баумана _____Смирнова Е.В._____

Руководитель от предприятия _____Юн С._____

Оглавление

Введение.....	3
1. Web-технологии Java Script, jQuery, jQuery Mobile, TAU, HTML, CSS.....	4
1.1 Разработка простейшего приложения для смартфона.....	4
1.2 Многостраничные и одностраничные приложения.....	5
2. СУБД IndexDB и WebSQL. Доступ к системным функциям.....	7
3. Приложения для смарт-часов.....	9
4. Файловая система Tizen.....	11
5. Индивидуальное задание: проект SpeedRead.....	12
Заключение.....	20
Список использованной литературы.....	21

Введение

Целью практической работы являлось изучение операционной системы OS Tizen, получение навыков работы с TizenAPI, а так же усовершенствование навыков разработки веб-приложений в SDK Tizen Studio на языке Javascript для мобильных устройств на платформе Tizen: смартфонов, смарт-часов и смарт-ТВ. При разработке приложений в том числе использовались: библиотеки JQuery и JQuery Mobile, а также HTML и CSS. Также в рамках практики были изучены некоторые принципы работы в больших компаниях, некоторые актуальные направления современной IT-индустрии: машинное обучение, использование смартфона как компьютера, мобильные платёжные системы и т.д..

Итоговый проект должен был продемонстрировать практическое применение полученных. Необходимо было в относительно короткий срок разработать приложение для любого из устройств на ОС Tizen и представить приложение на защите проекта.

Разработка приложений велась в SDK Tizen Studio, в качестве учебных пособий использованы материалы лекций и мастер-классов, документация с сайта TizenDevelopers (<https://developer.tizen.org/>), а также документация к основному стеку веб-технологий JS, HTML, jQuery и jQuery Mobile.

В ходе практики практики было разработано несколько приложений, но в настоящем отчёте будут приведены лишь некоторые, так как, описание проектной работы во многом охватывает изученный материал.

1. Web-технологии Java Script, jQuery, jQuery Mobile, TAU, HTML, CSS

1.1 Разработка простейшего приложения для смартфона

JavaScript, HTML, CSS — это совокупность технологий, которые используются для построения простейших веб-приложения. Но для упрощения разработки сложных современных проектов разработчиками используются фреймворки, например, jQuery. jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX. На основе jQuery и разрабатывалось большинство функций в учебных и итоговом проектах практики. А для создания интерфейса понятного пользователю и соответствующего современным стандартам дизайна использовались библиотеки TAU и JQueryMobile — стандартные WebUI библиотеки.

Для изучения принципов работы вышеописанных технологий было написано простейшее приложение с двумя кнопками, по нажатию на которые показывается размер экрана и появляется и исчезает картинка.

1.2 Многостраничные и одностраничные приложения

Современные приложения как правило сложные по своей структуре. Существует два основных метода разработки многофункциональных веб-приложений: разработка многостраничных и одностраничных приложений. Поэтому необходимо владеть технологией работы с многостраничными приложениями или с технологией расширяющей функциональность одной страницы (SPA — Single Page Application). Для изучения принципов создания многостраничного приложения и улучшения навыков владения Java Script, JQuery, HTML, CSS, было разработано учебное приложение, состоящее из нескольких страниц. Первая страница имеет две кнопки — одна из них

показывает текущее время, вторая осуществляет переход на следующую страницу. На рисунке 1 приведены скриншоты первой и второй страниц приложения. На обеих страницах есть основной контент и кнопки навигации по страницам приложения, а также кнопки отвечающие за логику страницы. Так на первой странице отображается текущее время по нажатии кнопки «Current time», а на второй — можно изменить цвет текста.

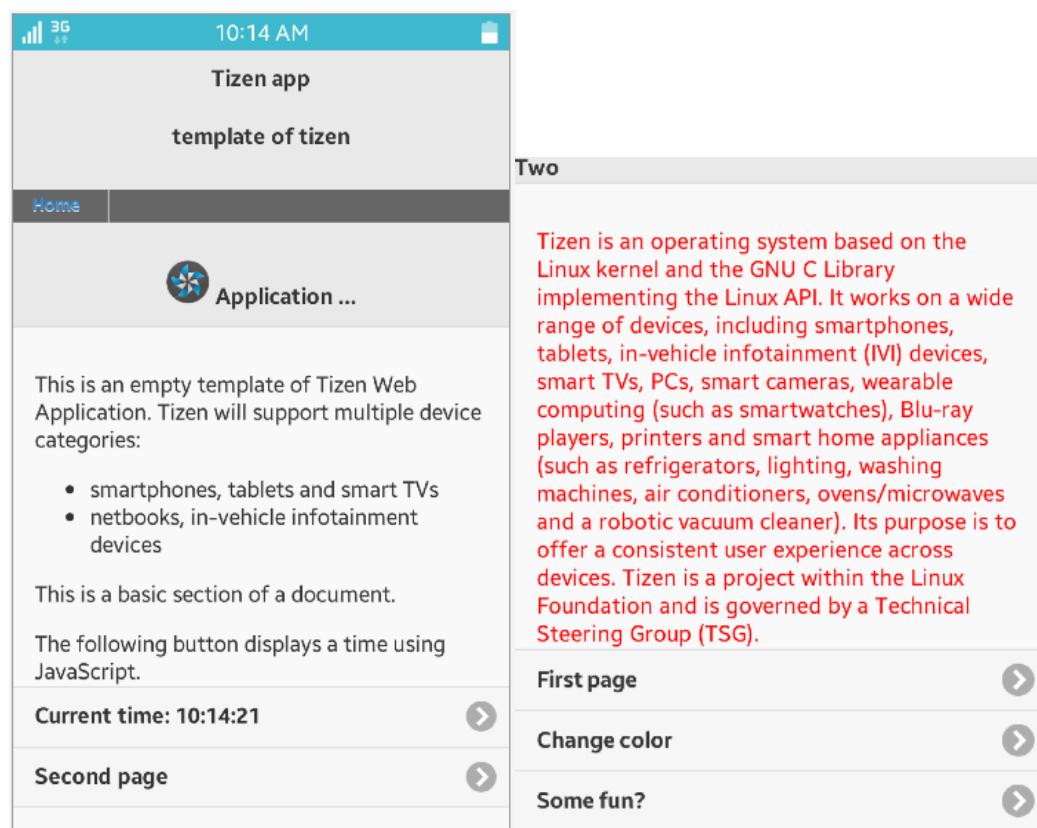


Рисунок 1 – Первая и вторая страница

Далее, была создана еще одна страница, также имеющая кнопки навигации на следующую и предыдущую страницы и содержащая элемент-изображение (``), который можно перетаскивать. Обработка перетаскивания была сделана с помощью возможностей jQuery.

Для создания наиболее интересных по дизайну или игровых приложений часто приходится использовать элемент типа `canvas`. Так последняя страница приложения имеет три элемента `canvas`: первый содержит в себе тексты различных цветов, размеров, шрифтов и положений на экране, второй заполнен рисунком по типу шахматной доски, а на третьем

реализовано анимированное изображение квадрата, перемещающееся по холсту.

В ходе выполнения практических задач, были изучены Java Script, jQuery, HTML, CSS, а также освоены методы создания мобильных приложений, использования Tizen API, применения различных тем оформления с использованием jQuery Mobile, TAU

2. СУБД IndexedDB и WebSQL. Доступ к системным функциям

Сложно представить современное приложение, которое бы не хранило никаких пользовательских данных, поэтому для создания полноценного приложения необходимо иметь навыки работы с СУБД, используемыми в веб-разработке. В ходе практики был получен опыт с двумя наиболее популярными API для клиентского хранилища: IndexedDB и WebSQL. Для изучения возможностей использования баз данных для хранения информации в устройствах с ОС Tizen, было разработано приложение, загружающее новостную ленту RSS с сайта 3DNews при помощи AJAX-запроса и сохраняющее загруженные статьи в базу данных для возможности оффлайн-просмотра новостной ленты. При наличии интернет-соединения новости из ленты отображаются на странице rss.html и сохраняются в базу данных, при отсутствии интернет-соединения AJAX запрос не может быть осуществлен и данные из базы данных выгружаются в новостную ленту на странице rss_offline.html. Для упрощения хранения изображений в базе данных был освоен метод преобразования изображения кодировкой Base64. То есть, в базе данных вместо изображения хранилась строка содержащая в Base64-код изображения. Для обратного преобразования хранящейся строки в изображение использовался элемент canvas, в который встроена функция преобразования Base64-кода в изображение. На рисунке ниже представлена страница с новостной лентой.

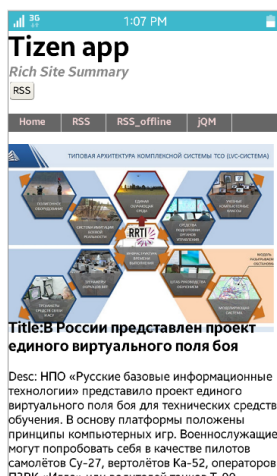


Рисунок 2 – Новостная лента, загруженная приложением

Особенностью разработки мобильных приложений является то, что разработчику нужно иметь доступ к системным функциям и параметрам. Для этого были получены навыки работы с Tizen API, которое позволяет без особых трудностей узнавать системные параметры устройств и управлять ими по Bluetooth и Wi-fi. В ходе практического занятия на еще одной странице приложения были решены задачи по управлению вибро-сигналом и bluetooth, а также получение данных о сети Wi-fi.

3. Приложения для смарт-часов

Для изучения методов разработки приложений для смарт-часов, было разработано приложение будильник. На рисунке 3 приведена стартовая страница приложения.



Рисунок 3 – Стартовая страница приложения будильник

При нажатии на надпись открывается окно установки времени срабатывания будильника; снимок экрана приведён на рисунке 4.

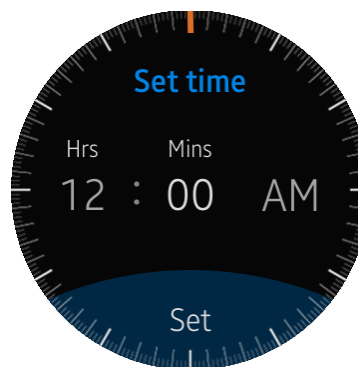


Рисунок 4 – Окно установки времени

При наступлении назначенного времени происходит пробуждение приложения из спящего режима, на экран выводится сообщение и устройство вибрирует до тех пор, пока пользователь не нажмет «ОК». Экран пробуждения представлен на рисунке 5.



Рисунок 5 – Экран пробуждения будильника

Таким образом, было разработано приложение-будильник, которого нет во встроенных приложениях смарт-часов. Для реализации данного приложения было необходимо использовать доступ к системным функциям устройства: поставить срабатывание будильника в расписании, вывести процессор спящего режима, включить экран и вибрацию.

4. Файловая система Tizen

С файловой системой Tizen позволяет в полной мере работать Tizen Filesystem API. Это отличает разработку веб-приложения под мобильную платформу от разработки браузерных веб-приложений, ведь в целях безопасности средствами простого Javascript нельзя писать в пользовательские директории на компьютере. Для получения практических навыков работы с файлами в Tizen OS был расширен функционал пункта 2, а именно, добавлена возможность сохранения файлов. В листинге 1 приведён исходный код функции сохранения файлов.

Листинг 1:

```
function writeTwoFiles(){

    var documentsDir;
    var dir = tizen.filesystem;
    var name = "";
    function onSuccess(files) {

        for(var i = 0; i < files.length; i++){
            name+=files[i].name;
            name+=", "
        }
        alert("Files names are " + name);
        var dir1 = documentsDir.createDirectory("gg");
        var file1 = dir1.createFile("wi-fi");
        if (file1 != null) {
            file1.openStream(
                "w",
                function(fs1){
                    var text = document.getElementById("div_wifi").innerHTML;
                    fs1.write(text);
                    fs1.close();
                }, function(e1){
                    console.log("Error " + e1.message);
                }, "UTF-8"
            );
        }

        var file2 = documentsDir.createFile("input");
        if (file2 != null) {
            file2.openStream(
                "w",
                function(fs2){
                    fs2.write("");
                    fs2.close();
                }, function(e2){
                    console.log("Error " + e2.message);
                }, "UTF-8"
            );
        }
        getWifiInfo();
    }
}
```

5. Индивидуальное задание: проект SpeedRead

В ходе выполнения итогового проекта, индивидуальная тема которого заранее обсуждалась с преподавателем, было разработано приложение для смартфона SpeedRead — тренажер скорочтения. Целью проекта было создание обучающего приложения, позволяющего пользователю улучшить навыки скорочтения, что позволит ему научиться быстрее воспринимать и понимать текстовую информацию. Скорость прочтения книг после использования приложения может возрасти в среднем в 2-3 раза. Также тренажер будет полезным как взрослым, так и детям. Проект создан с использованием следующих технологий: Javascript, jQuery, jQuery Mobile, языки вёрстки HTML и CSS. Также использовались API Tizen: Power API и Filesystem API. Полное техническое задание и исходные коды представлены в листе задания и приложении 1 соответственно.

Приложение является многостраничным, главная страница — библиотека (рисунок 6). Пользователь может выбрать книгу или загрузить новую, так же пользователь видит свою максимальную скорость чтения и может сбросить результат, нажав на кнопку «Reset».

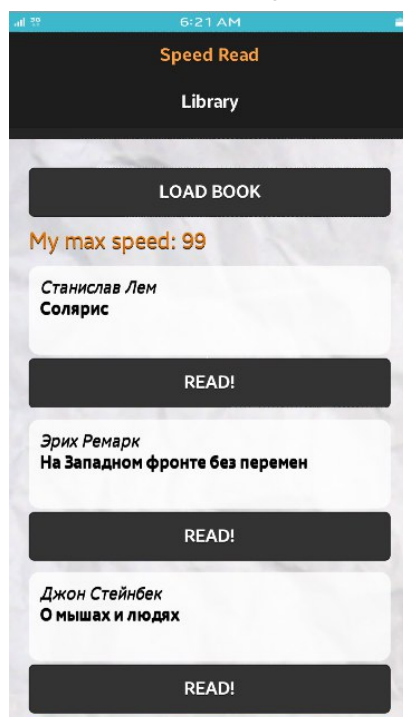


Рисунок 6 — Страница библиотеки

При переходе со страницы библиотеки на страницу загрузки книги (load.html) пользователю предоставлялся интерфейс, представленный на рисунке 7.

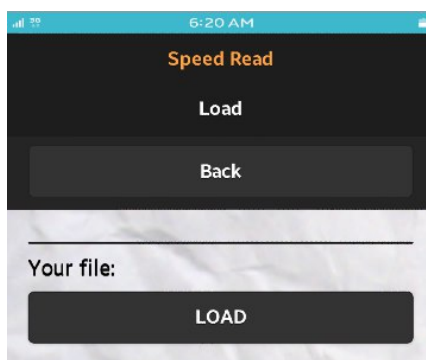


Рисунок 7 — Страница загрузки книг в библиотеку

Для загрузки книги в библиотеку следующим образом использовалось Filesystem API. В элемент input типа file загружался файл из любого пользовательского каталога устройства, причем было наложено ограничение на тип файла (только fb2 — формат электронной книги). При помощи средств Javascript и Filesystem API содержимое input перезаписывалось в файл с тем же именем, но уже в рабочем каталоге приложения. Копирования файла в определенный каталог было сделано для удобства пользователя и разработчика. Листинг 2 - листинг файла load.js с функциями работающими с файлами.

Листинг 2:

```
var name; //имя добавляемого файла
var ansString; //содержимое этого файла

function writeTestFile(){ ///////////////NAME, ANSSTRING////////////////////

    var documentsDir;
    var dir = tizen.filesystem;
    var flag = 0;

    function onSuccess(files) {
        var format = name.split(".");
        if (format[1]=== "fb2"){
            for(var i = 0; i < files.length; i++) {
                console.log("File Name is " + files[i].name); // displays file name
                if (files[i].name===name){
                    flag = 33;
                }
            }
        }
    }
}
```

```

    }
}

console.log("после вывода списка файлов");
("list_of_files").innerHTML = str + '<br>';

console.log("перенос строки после вывода списка файлов");
console.log(flag);

if (flag===33) {
    reWriteF();
}
else {
    newWriteF();
}

function newWriteF(){
    console.log("newWriteF");
    console.log(name);
    var testFile = documentsDir.createFile(name);
    console.log("FILE = " + testFile);
    testFile.openStream(
        "w",
        function(fs){

fs.write(ansString);////////////////////////////////////////
        fs.close();
        alert("Book successfully uploaded");

document.getElementById("restart").click();////////////////////////////////
        }, function(e){
            console.log("Error " + e.message);
            }, "UTF-8"
        );
        console.log("newWriteF");
        /*}*/
    }
    function reWriteF() {
        alert("File already exists");
        console.log("File already exists");
    }
}
else {
    alert("Please, choose .FB2 book");
    console.log("File already exists");
}
}

function onerror(error) {
    console.log("The error " + error.message + " occurred when listing the files in the
selected folder");
}

tizen.filesystem.resolve(
    'documents',
    function(dir){
        documentsDir = dir;
        dir.listFiles(onsuccess,onerror);
    }, function(e) {
        console.log("Error" + e.message);
    }, "rw"
);
}

function loadFile(files) {
    var file = files[0];
    document.getElementById("file_name").innerHTML = "Your file: ";

```

```

        document.getElementById("file_name").innerHTML =
document.getElementById("file_name").innerHTML + file.name; //чтобы скрыть оригинальный
инпут
        var myReader = new FileReader();
        ansString = "";

        myReader.readAsText(file);
        myReader.onload = function(e) {
            ansString = e.target.result;
            name = file.name;
            writeTestFile();
        }
    }

function newDoc() {
    window.location.assign("index.html");
}

```

Если, находясь на странице библиотеки, пользователь нажмет на кнопку «Read» под нужной ему книгой, то он перейдет на страницу чтения книги (read.html), скриншот которой представлен на рисунке 8.

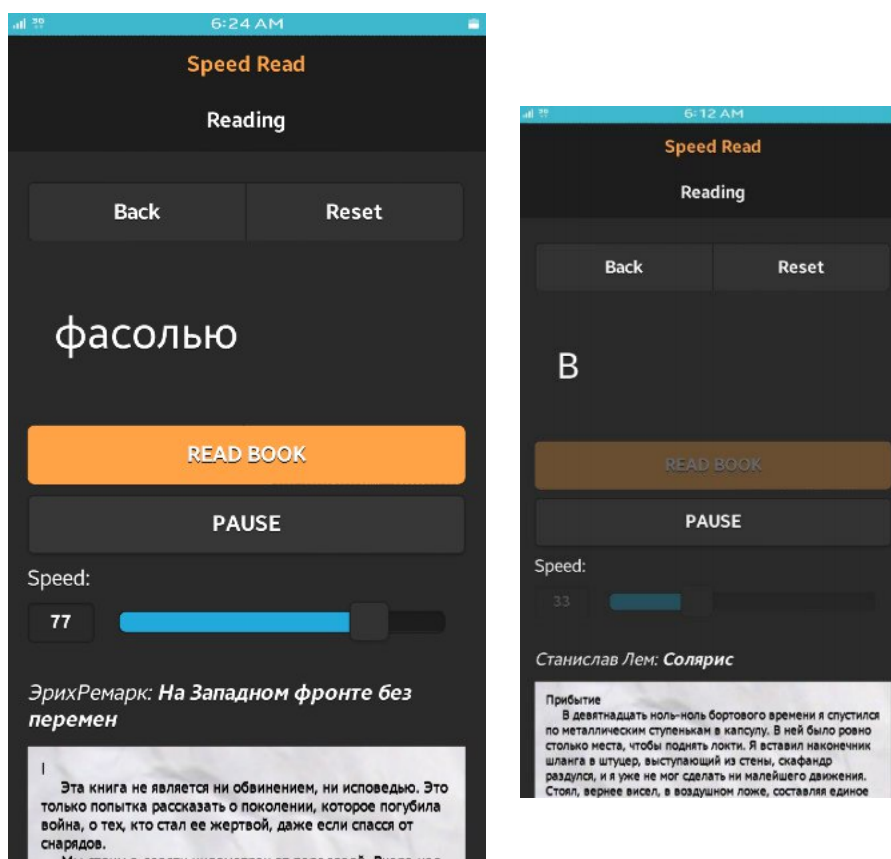


Рисунок 8 — Страница чтения книги в режиме паузы и в режиме чтения

Как видно из скриншотов, для начала чтения пользователю необходимо нажать на кнопку «Read book», а до этого в режиме паузы пользователь может настроить скорость чтения при помощи ползунка «Speed». Основной задачей при разработке являлась задача парсинга xml-файла формата fb2. Нужно было извлечь автора книги и ее текст по абзацам. Листинг логики, связанной с парсингом xml-файла, представлен ниже.

Листинг 3:

```
function getXmlDoc(cont){
    document.getElementById("auth").innerHTML = "";
    var myText = cont;

    parser = new DOMParser();
    xmlDoc = parser.parseFromString(myText,"text/xml");

    return xmlDoc;
}

function getAuth(text) {
    var cont = text;
    var xmlDoc = getXmlDoc(cont);
    $(xmlDoc).find("title-info").find("author").each(function () {
        $("#auth").append($(this).find("first-name").text() + " " + $(this).find("last-name").text() + ": ");
    });

    $(xmlDoc).find("title-info").each(function () {
        $("#auth").append("<b>" + $(this).find("book-title").text() + "</b>");
    });
}

function getText(text) {
    var cont = text;
    var xmlDoc = getXmlDoc(cont);

    var arrOfPar = [];

    document.getElementById("book_cont").hidden = false;

    $(xmlDoc).find("section").find("p").each(function () {
        $("#book_cont").append($(this).text()+"<br>&nbsp; &nbsp; &nbsp;");

        arrOfPar[arrOfPar.length] = $(this).text(); //массив параграфов
    });

    var path = localStorage.getItem("curent");
    var indexWord = localStorage.getItem(path).split("_"); // установили там, где остановились

    indexP = indexWord[0];
    indexInP = indexWord[1];
    sizeP = indexWord[2];
}
```



```

var numPar = arrOfPar.length;

var slider = document.getElementById("slider-fill").value;
var record = localStorage.getItem("record");

if (slider > record) { // если побит рекорд, то изменяем
    localStorage.setItem("record",slider);
}

var speed =((100-slider)*15 + 1);

interval = setInterval(printOneWord,speed,arrOfPar,numPar);
}

```

Особенностью данной страницы с точки зрения разработки под смартфон является то, что пользователь после начала чтения не взаимодействует с устройством и оно переходит в режим ожидания и экран отключается, чтобы устройство постоянно находилось в активном режиме применено Power API (листинг 5).

Листинг 5:

```

function readBookFile(){ // здесь и проверяем есть ли книга в локалсторадже

    tizen.power.request('CPU', 'CPU_AWAKE');
    tizen.power.request('SCREEN', 'SCREEN_NORMAL');
    tizen.power.turnScreenOn();
    ...
}

```

При возвращении пользователя на страницу библиотеки, выхода из приложения или нажатия на кнопку «Pause» приложение сохраняет скорость если она превысила рекордную и слово, на котором остановилось чтение, в localStorage (листинг 6).

Листинг 6:

```

var flagIn = 0;
var newName = '';

function readBookFile(){ // здесь и проверяем есть ли книга в локалсторадже

    tizen.power.request('CPU', 'CPU_AWAKE');
    tizen.power.request('SCREEN', 'SCREEN_NORMAL');
    tizen.power.turnScreenOn();

    var path = localStorage.getItem("curent");

    if (isInLocalStorage(path)===true) { // если книга встртилась в localStorage,
то увеличим flag

```

```

        flagIn = 33;
        newName = path;
    }
    else { // иначе в локалсторадж
        localStorage.setItem(path,"0_0_0");
    }

    document.getElementById("read_book").disabled = true;

    $( "#slider-fill" ).slider({
        disabled: true
    });

    tizen.filesystem.resolve("documents", function(dir)
    {
        file = dir.resolve(path);
        file.openStream(
            "r",
            function(fs) {
                var text = fs.read(file.fileSize);
                fs.close();

                getText(text);
                getAuth(text);

            }, function(e) {
                console.log("Error " + e.message);
                book = document.getElementById("book_err");
                book.innerHTML = e.message;
            }, "UTF-8");
    });
    console.log("OK BOOK");
    document.getElementById("read_book").disabled = true;
}

```

Также была написана функция, с заданной пользователем скоростью отображающая по одному слову на экране пользователя (листинг 7)

Листинг 7:

```

var indexP = 0;
var indexInP = 0;
var index = 0;
var sizeP = 0;

function printOneWord(arrOfPar,numPar){
    console.log(indexP + '=>' + numPar);
    if (indexP < numPar) { // если параграфы не закончились, то выводим
текущее слово нужного
        var parCont = arrOfPar[indexP]; // текст в параграфе
        var arrWordsInP = parCont.split(" "); // массив слов в параграфе
        sizeP = numPar;
        if(indexInP < arrWordsInP.length) { // если пробежались не по всем
словам, то выводим следующее
            document.getElementById("box").innerHTML = arrWordsInP[indexInP];
            indexInP++;
        }
        else { // иначе переходим к следующему параграфу

```

```
        indexP++;  
        indexInP = 0;  
    }  
}  
else { // чистим интервал  
    clearInterval(interval);  
}  
}
```

Таким образом, в результате работы над проектом было разработано полностью функционирующее приложение, соответствующее техническому заданию. Проведено тестирование приложения SpeedRead как в симуляторе, так и на реальном устройстве. В ходе тестирования ошибок выявлено не было.

Заключение

В ходе практики были приобретены навыки работы с OS Tizen и TizenAPI и усовершенствованы навыки разработки веб-приложений, в том числе с использованием js-фреймворка jQuery. Также были изучены библиотеки jQuery Mobile, TAU, языки вёрстки HTML и CSS и освоены некоторые современные методы разработки приложений. Помимо этого в ходе прохождения практики были изучены некоторые принципы работы в больших компаниях, освоены методы рефакторинга кода и его быстроедействие и получены знания о аппаратной платформе Samsung.

Список использованной литературы

1. Лекционные материалы
2. <https://developer.tizen.org/>
3. <http://developer.samsung.com/>