“GOODEED” Cordova-based mobile application

Project by Manuel Gajo, gajo.manuel@gmail.com

Abstract

Goodeed is a social entrepreneurial project whose business and design parts have developed during *Masterlab in Digital Economics & Entrepreneurship*. The project is still evolving and will become soon a company registered in Italy as *social enterprise*.

Goodeed has the aim to innovate online donations world:

- *for the donor*: by making donations traceable, project-centric (each project has a “story” of how each object is used) and well documented
- *for the association*: by proving a powerful means of communication and “free” advertising
- *for big companies*: by providing them a “showcase” for their social corporate responsibility needs

Project description

The project will be composed of:

- *a web site* to serve as a showcase for projects and big companies
- *a mobile application* to allow associations to easily document their projects and as a means to make donors involved wherever they are.

The *mobile application* works in this way (as shown in *User interaction schema* paragraph):

- The user is logged either as association or donor
- The association chooses one of their projects and document it by
    - taking pictures
    - recording audio clips
    - choosing video clips (from the Internet)
    - posting updates (title and description)
- Both the association and the user have access to project’s diary (donors have access to the projects they are supporting by donating, associations have access to their own projects)

Implementation details

This demo has the primary aim to serve as MVP (minimum viable product) for business idea’s testing, so the purpose of presenting it as *Multimedia Systems* final project is to show *mobile framework and user interface implementation* as well as *Cordova’s and HTML5’s multimedia capabilities*.

Technical implementation is based on Cordova framework and on Javascript, HTML5 and CSS languages. Data shown is simulated and there is not a true backend. However content added is stored into either the device or a remote server, and a basic textual database is used.

A contextual menu (different for the user and the association) is available. Only project’s diary (and *starred projects* page) option is available at this time, since it is the part of the project which best demonstrates *mobile implementation* and *HTML5’s multimedia capabilities*.

CSS classes are designed for mobile by using *em* and *%* values.
Also *viewport* metatag is used to assure correct output on different screen sizes.

One project diary is available and is the same for both the donor and the association. Project description is static but content added by the user (shown in the second page) is dynamically loaded.

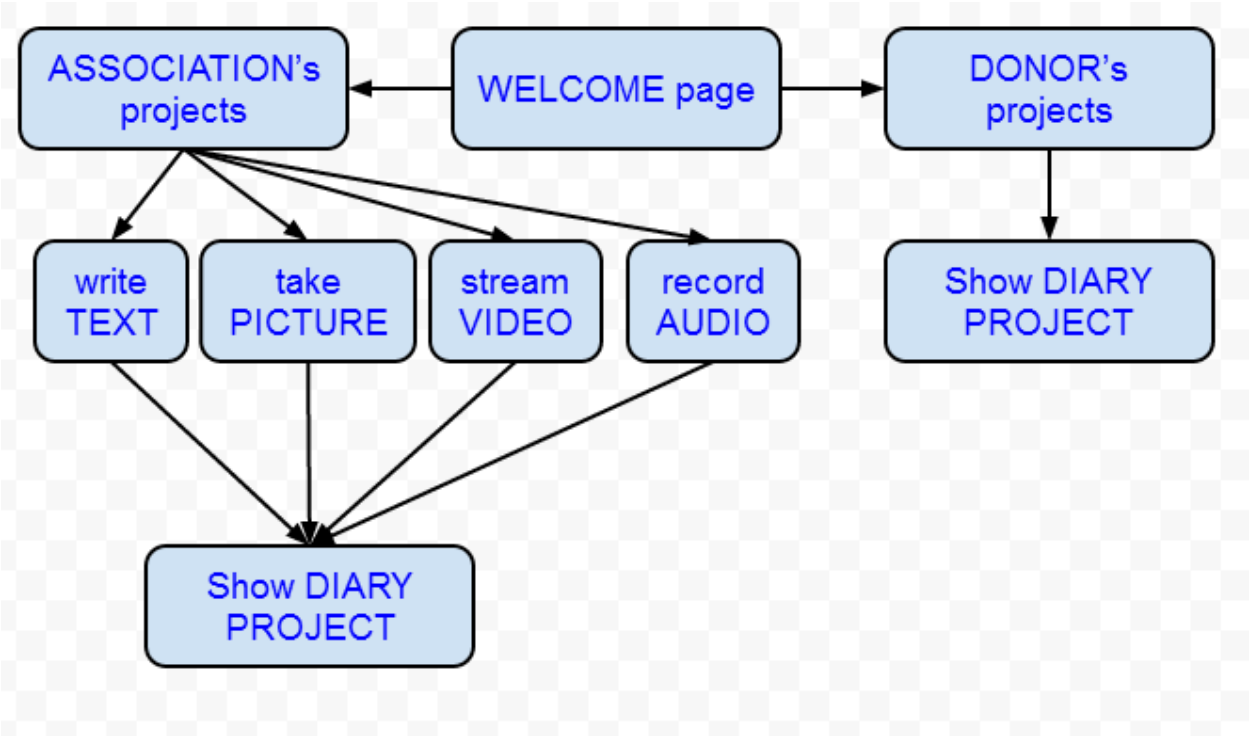
Due to *Cordova for Android* limitations, it is not possible to capture video (not even with the code provided in *Cordova media plugin* wiki: https://github.com/apache/cordova-plugin-media/blob/master/doc/index.md), hence it is possible to insert one or more source (in case of different video formats) from the Internet.
The following sample videos can be used:

- http://myownlibrary.altervista.org/goodeed_mvp/videos/sample_video.mp4
- http://myownlibrary.altervista.org/goodeed_mvp/videos/sample_video.ogg
- http://myownlibrary.altervista.org/goodeed_mvp/videos/sample_video.webm
- http://myownlibrary.altervista.org/goodeed_mvp/videos/sample_video.avi
- http://myownlibrary.altervista.org/goodeed_mvp/videos/sample_video.mov

Please note that for these limitations some video formats/codecs may not play correctly. Also MOV is a format not supported natively by HTML 5. *jwplayer* may solve issue with the codes and with MOV container. In general *mp4* is playable and should be enough as source for a video in a cordova-based application.
Also, it is possible to add a *.vtt* source to display *subtitle*.

Back button handler has been implemented to execute suitable action (refer to *backButtonHandler()* function for details)

User interaction schema



Tools and languages used

The project has been successfully tested on *LG Nexus 5* Android device with the following technical specifications: Android 4.4.2, 4.95 inches display, Full HD resolution (1920x1080), Qualcomm Snapdragon 800 CPU, Adreno 330 GPU.

Cordova version used to compile the project is 3.5.0.

List of tools and library used:

- Java JDK 8 64-bit on Windows 8.1 64-bit.
- Android SDK with platform tools and build targets (19 e and 20)
- Eclipse Kepler (Standard) IDE 64-bit
- ADT (Android Development Tools) plugin for Eclipse
- Apache ANT 1.9.4
- NodeJS v0.10.29
- Git for Windows 1.9.4
- Plugman for Cordova: to install plugins
- Cordova's core plugins: media, camera, file
- Cordova's third-party plugins: "set screen orientation", "display toast notifications"
- *Wifi ADB* Android application, to deploy the application over WiFi
- Javascript plugins and frameworks:
    - *JS Dynamic framework* by Manuel Gajo (released under GNU v3 open source license)
    - jQuery v1.7.1

    - blockUI.js : block/unblock whole page / specific page
    - spin.js: show visually when a page is loading
- CSS “normalize” project, to improve cross-browser compatibility, when not deployed within cordova
- Bootstrap framework (v3.1.1), for responsive mobile design

From web files (html,js,css etc.) to **Android’s apk** : **step-by-step tutorial**

- Download and install Java JDK 8 64-bit and set *system environmental* variables
    - add to PATH: JDK’s */bin/* subdirectory
    - set JAVA_HOME: JDK’s *root* directory
- Download and install *Eclipse Kepler (standard) 64-bit* *http://eclipse.org/downloads/packages/eclipse-standard-432/keplersr2*
- Download and install *standalone Android SDK*: http://developer.android.com/sdk/index.html#download
    - From Android SDK GUI (open it from command line by executing *android* command), install *Tools* , *Platform tools* and *Android SDK Build Tools* (both versions 19 and 20)
    - set *system environmental* variables
        - add to PATH: *platform-tools* and *tools* folders
        - set ANDROID_HOME: SDK’s *root* directory
- Download ANT (no need to install),unzip it and set *system environmental* variables
    - add to PATH: ANT’s */bin* subdirectory
    - ANT_HOME: ANT’s *root* directory
- Install NodeJS: http://nodejs.org/
- Install *Git for Windows*: http://msysgit.github.io/ (“Use from command line” installation)
- Instal and configure ADT plugin in Eclipse:
    - Tutorial: http://developer.android.com/sdk/installing/installing-adt.html
- Download zip version (not CLI) of Apache Cordova (https://www.apache.org/dist/cordova/platforms/) latest version (3.5.0, Android) and unzip it

- Open a new command line (*Run → cmd*)
    - Go to cordova's */bin/* directory in command line
    - Create new project by executing: *create Goodeed com.madee.goodeed Goodeed*
    - Go to cordova's */framework* directory and execute the following commands to generate *cordova-3.5.0.jar*
        - *android update project -p . -t android-19 --subprojects*
        - *ant jar*
- *In Eclipse: File → Import.. → Existing Android Code into workspace*
    - Choose directory *goodeed* created by Cordova
- Delete all files and folders (except *cordova.js*) from */assets/www* project's folder
- Copy *web_files* folder content into */assets/www* project's folder
- Overwrite project's *AndroidManifest.xml* file with provided one (additional *permissions* and few more needed lines have been added)
- Add empty */libs/* folder in project's root folder
    - Copy *cordova-3.5.0.jar* file generated previously into this folder
    - Add *cordova-3.5.0.jar* to Eclipse's build path
- Open a new command line (*Run → cmd*)
    - Install plugman: *npm install -g plugman*
    - Go to *project's parent directory* directory
    - Install needed cordova's *core* plugins using plugman
        - plugman install --platform android --project Goodeed --plugin *org.apache.cordova.camera*
        - plugman install --platform android --project Goodeed --plugin *org.apache.cordova.file*
        - plugman install --platform android --project Goodeed --plugin *org.apache.cordova.media*
    - Install needed cordova's *third-party* plugins using plugman
        - plugman install --platform android --project Goodeed --plugin *https://github.com/gitawego/pg-plugin-screen-orientation.git*
        - plugman install --platform android --project Goodeed --plugin *https://github.com/EddyVerbruggen/Toast-PhoneGap-Plugin.git*
- Apply the following changes to auto-created *java* file in *com.madee.goodeed* package
    - Refactor class name by renaming it to *MainActivity*
- Connect to *virtual device* or *real device*
    - In case of *real device*: connect Android device via usb cable
        - In case of *rooted device* it is possible to use *Adb over wifi*
            - install *Wifi ADB* application: https://play.google.com/store/apps/details?id=com.ttxapps.wifiadb
            - enable *Wifi ADB* (device must be connected to WiFi)
            - in Windows, from command line, run *adb connect <device IP>*
    - In case of virtual device install *Device drivers for Windows* and follow tutorial at: http://developer.android.com/tools/help/avd-manager.html
- In Eclipse: deploy *Android application*