

Projet d'adaption IHM

Réalisé par :

Insaf Jebara

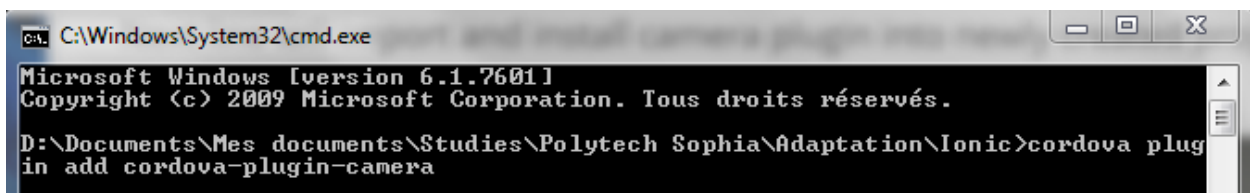
Figure 1 : code source camera	3
Figure 2 : Controleur video.....	4
Figure 3 : Capture vidéo	5
Figure 4 : Liste de cours.....	5
Figure 5 : cours Ionic	6
Figure 6 : code source cours Ionic.....	6
Figure 7 : Controleur photo.....	7
Figure 8 : capture photo.....	7

Annexe de tutoriel Ionic

Etant donné que je n'ai pas pu aboutir à une application complètement fonctionnelle avant la date limite d'envoi du rapport. Je me permets de mettre en annexe la partie d'accès à la caméra permettant d'ajouter une photo pour un cours précis ou bien l'ajout d'une vidéo après s'être authentifié à l'application.

Après authentification nous avons accès à cette interface. A l'appui sur le bouton caméra, nous avons accès à la caméra du smartphone.

Avant de commencer le développement, il y a un plugin à installer, celui de cordova.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

D:\Documents\Mes documents\Studies\Polytech Sophia\Adaptation\Ionic>cordova plugin
add cordova-plugin-camera
```

Ensuite pour développer l'interface, nous avons créé une div qui va contenir le clip filmé :

```
<ion-view view-title="Upload new courses" ng-controller="VideoCtrl">
  <ion-content>
    <div>
      <p>Upload a new video course</p>
    </div>
    <br/>
    <button class="button button-full button-calm" ng-click = "captureVideo()" >
      Camera
    </button>
    <div>
      <div class="list">
        <div class="item item-divider">
          <i class="ion-videocamera"></i>
          LIVE
        </div>
        
      </div>
    </div>
  </ion-content>
</ion-view>
```

Figure 1 : code source camera

Le bouton camera nous renvoi à la fonction captureVideo() qu'on a défini dans le contrôleur.

Ce bout de code ne permet pas tout seul d'accéder à la caméra donc nous avons dû ajouter un code au contrôleur pour implémenter les fonctions définies :

```

.controller('VideoCtrl', function($scope, $cordovaCapture, VideoService) {
    $scope.clip = '';

    $scope.captureVideo = function() {
        $cordovaCapture.captureVideo().then(function(videoData) {
            VideoService.saveVideo(videoData).success(function(data) {
                $scope.clip = data;
                $scope.$apply();
            }).error(function(data) {
                console.log('ERROR: ' + data);
            });
        });
    };

    $scope.urlForClipThumb = function(clipUrl) {
        var name = clipUrl.substr(clipUrl.lastIndexOf('/') + 1);
        var trueOrigin = cordova.file.dataDirectory + name;
        var sliced = trueOrigin.slice(0, -4);
        return sliced + '.png';
    }

    $scope.showClip = function(clip) {
        console.log('show clip: ' + clip);
    }
})

```

Figure 2 : Controleur video

Comme précisé auparavant, ionic permet l'utilisation d'éléments prédéfinis. Ici, nous pouvons voir que nous avons \$cordovaCapture, une fonction du plugin qui fait appel à notre fonction définie dans le bouton html.

\$scope est une fonction prédéfinie de Angular JS qui permet de faire la liaison entre le contrôleur et la vue. On y stocke presque tout pour pouvoir le réutiliser.

La deuxième partie de ce code permet la récupération de la capture vidéo et son découpage pour enfin être affichée en tant qu'image d'extension .png

L'interface de capture vidéo :

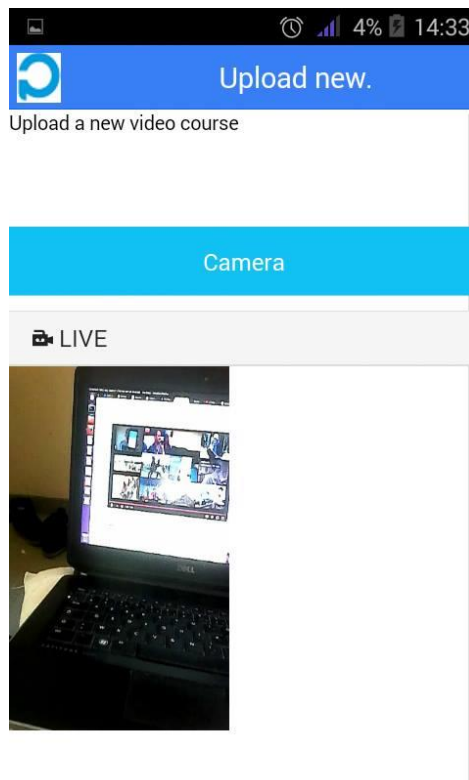


Figure 3 : Capture vidéo

Accès à l'appareil photo :

Nous avons choisi de représenter chaque cours par un bouton simple (<button>). A l'appui sur le bouton Ionic par exemple nous accédons à l'interface du cours où l'internaute/utilisateur peut ajouter des images du cours ou des capture écran.

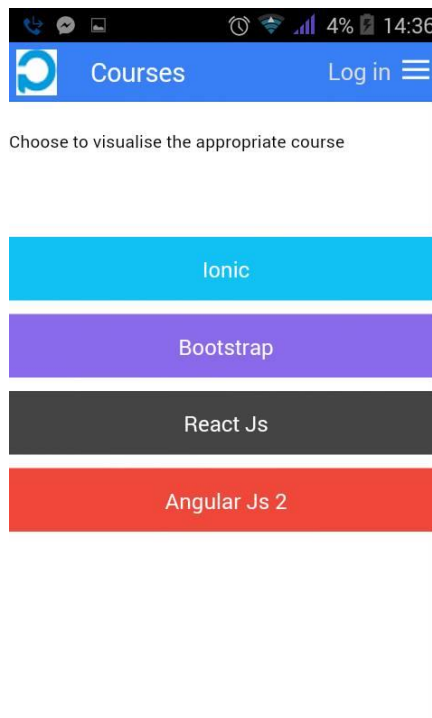


Figure 4 : Liste de cours

Pour choisir Ionic, nous avons cette interface qui s'affiche :



Figure 5 : cours Ionic

```
<ion-view view-title="Ionic Course" ng-controller="PictureCtrl">
  <ion-content>
    
    <div>
      Ionic est un framework permettant de developper des applications hybrides
      Il utilise principalement des technologies web et permet de deployer des applications natives
    </div>
  </ion-content>
  <ion-nav-footer>
    <div class="tabs tabs-icon-left">
      <a class="tab-item">
        <i class="icon ion-camera" ng-click="takePhoto()"></i>
        Camera
      </a>
    </div>
  </ion-nav-footer>
</ion-view>
```

Figure 6 : code source cours Ionic

Comme pour la vidéo, l'implémentation de chaque fonction définie dans html se fait dans le contrôleur.

```

.controller('PictureCtrl', function ($scope, $cordovaCamera) {

    $scope.takePhoto = function () {
        var options = {
            quality: 80,
            destinationType: Camera.DestinationType.DATA_URL,
            sourceType: Camera.PictureSourceType.CAMERA,
            allowEdit: true,
            encodingType: Camera.EncodingType.JPEG,
            targetWidth: 300,
            targetHeight: 300,
            popoverOptions: CameraPopoverOptions,
            saveToPhotoAlbum: false
        };

        $cordovaCamera.getPicture(options).then(function (imageData) {
            $scope.srcImage = "data:image/jpeg;base64," + imageData;
        }, function (err) {
            // An error occurred. Show a message to the user
        });
    }
});

```

Figure 7 : Controleur photo

CordovaCamera.getPicture permet ici d'ouvrir l'interface de récupération d'image dans l'interface.



Figure 8 : capture photo