What is the main goal of the feature?

The main purpose of this feature is to allow Acted customers to display an artist profile page and to allow the artist to modify and customize his profile page.  
Consequently, this page will appear and behave differently, depending on the user who is loading the page.

What are the feature inputs?

* http://www.acted.co/profile/[ARTIST\_NAME] URL.
* The design mockup of the profile page, under the “profile.html” file, in the
* Mockup repo. Acted database.

What are the feature outputs?

A dynamically generated HTML document, respecting the “profile.html” file design, filled with the [ARTIST\_NAME] artist information (stored in the database).

What are the main business rules implemented?

We have implemented min/max price calculation and Artist's rating.

Define a quick “How it works” schema.

URL **http://[SITENAME]//profile/[ARTIST-NAME]** handles ProfileController::showAction. Action extracts the Artist data through Doctrine EntityManager (including Performances and Feedbacks) and renders ActedLegalDocsBundle:Profile:show.html.twig. We use some Form Type for adding/editing Artist’s information, e.g. ArtistType, MediaUploadType, PerformanceType, UserType, etc. We use jQuery AJAX for editable fields.

Which bundle handles the feature?

ActedLegalDocsBundle contains all application logic.

What are the main classes?

Acted\LegalDocsBundle\Entity\Artist - provides all useful date for Artist.

Acted\LegalDocsBundle\Model\MediaManager - contains methods for images uploading and extracting meta info for video.

Acted\LegalDocsBundle\HttpFoundation\File\Base64File - convert base64 to File object.

Acted\LegalDocsBundle\Form\ArtistType - main Form for Artist.

Acted\LegalDocsBundle\Form\MediaUploadType - provides image uploading and Gallery validation.

Acted\LegalDocsBundle\DataFixtures\ORM\LoadArtistData - provides fixtures for Artist.

Acted\LegalDocsBundle\Controller\ProfileController - contains actions for showing and editing artist profile.

What are the services?

gedmo.listener.tree - provides Nested Sets for categories.

app.twig\_extension - contains widget for categories output.

What are the Twig templates?

ActedLegalDocsBundle:Default:header.html.twig - header;

ActedLegalDocsBundle:Profile:userSection.html.twig - user header (name, avatar, background image);

ActedLegalDocsBundle:Profile:ordersSection.html.twig - performances list;

ActedLegalDocsBundle:Profile:biographySection.html.twig - biography block;

ActedLegalDocsBundle:Profile:feedbacksSection.html.twig - feedbacks list;

ActedLegalDocsBundle:Profile:mediaSection.html.twig - user gallery;

ActedLegalDocsBundle:Profile:feedsSection.html.twig - Facebook feed;

ActedLegalDocsBundle:Default:footer.html.twig - footer;

ActedLegalDocsBundle:Profile:modals.html.twig - popups;

ActedLegalDocsBundle:Default:pagination.html.twig - paginator;

What are the add-on bundle you installed?

LiipImagineBundle provides image scaling. JMSSerializerBundle is used for Entity serialization (responses for AJAX). KnpPaginatorBundle - provides flexible pagination. Embed/Embed is used for extracting meta information from third party URLs (Vimeo, Youtube).

What are the commands to run to install the feature?

$ composer update

$ php app/console doctrine:schema:drop --force

$ php app/console doctrine:schema:create

$ php app/console doctrine:fixtures:load

$ php app/console doctrine:migrations:migrate

What are the commands to run to test the feature?

We do not have tests for this feature.

How to test the feature?  
Go to profile list page - **http://[SITENAME]/profile/list**. You will see all artists. Click artist name and you will redirect to artist page - **http://[SITENAME]//profile/[ARTIST-NAME]**. This is a read-only page. If you want to see editable page, you should append URL with user ID, e.g. **http://[SITENAME]//profile/[ARTIST-NAME]?user=125**.