Product Backlog Item: 18285

Manage builder logos in VEO Admin

# Story

## Description

Given that we are getting more branding work to enable the sales effort, we often get blocked on the theme by lack of data or because we are dependent on the Echelon logo. Intent: be able to manage builder logos in VEO Admin rather than relying on Echelon logos.

## Acceptance Criteria

Given that a user has the permission to manage organizations (or manage themes, depending on implementation):

* The user can view logo currently assigned to organization.
* The user can upload a builder logo.
* Given that a logo has been updated and associated with organization:
  + The logo appears in the usual place in the top left hand of the header
  + The logo appears in the widget used to change the current organization
  + The logo appears on the selections report
  + The logo appears on the dmh spending summary report
* The uploaded logo should be no larger than [xkb -tbd]
  + The ui provides feedback and prevents exceeding this limit
* Write a script to grab existing production logos and insert them into our builder logo structure
* If no image exists for an organization
  + No image appears

# Code Changes

Within the database, we can utilize the SQL table VEOSolutionsSecurity.dbo.organization\_images and its entity class OrganizationImage.

Update to OrganizationImageCategory for a new enum of BUILDER\_LOGO.

The following UseCases will need to be created:

1. InsertOrganizationImage.cs  
   Will create a new record of OrganizationImage.  
   Configure on OrganizationId, Category, FileType and ImageData.
2. UpdateOrganizationImage.cs  
   Will update an existing record of OrganizationImage.  
   Configure on Id, Filetype and ImageData.  
   Updates FileType and ImageData.
3. DeleteOrganizationImage.cs  
   Will delete an existing record of OrganizationImage.  
   Configure on OrganizationImage.Id.
4. GetAllOrganizationImages.cs  
   Will get all OrganizationImage for an OrganizationID.  
   Configure on OrganizationId.
5. GetOrganizationImageCategories.cs  
   Will get a list of Categories of images.

To move the reliance of Organization Logos from the old way to the new way, the following updates need to be addressed:

* Obsolete the usage of OrganizationDomainRepository.GetBuilderLogo.
* Update GetWishListReport.BuildHeaderDTO method to fetching the logo through the GetOrganizationImage UseCase. Also, move the inversion logic into the use case.
* Update GetBuilderLogo UseCase to utilize fetching the logo through the GetOrganizationImage UseCase. Also, move the inversion logic into the use case.

New endpoints created for fetching OrganizationImages:  
Update the class ImageController with the following:

* GET api/Image/organizations/{organizationId}/{ImageCategory}
  + Get a specific OrganizationImage on OrganizationID and OrganizationImageCategory.
  + Fully expose the GetOrganizationImage Use case to fetch any image configured.
* GET api/Image/organizations/{organizationId}
  + Get all OrganizationImages on OrganizationID.
  + Utilize GetAllOrganizationImage.cs
* POST api/Image/organizations/{organizationId}/{ImageCategory}
  + Body: { imageData: <binary Image data>, imageType < supported image types> }
  + imageType => Look at what OrganizationImage supports at this time.
  + Utilize the InsertOrganizationImage.cs
* PUT api/Image/organizations/{Id}
  + Body: { imageData: <binary image data>, imageType <supported image types> }
  + Utilize UpdateOrganizationImage use case
* DELETE api/Image/organizations/{Id}
  + Deletes OrganizationImage on Id.
  + Utilize DeleteOrganizationImage use case

To support the maintenance of organization images in VEO Admin a new tab will be created on the viewOrganization view. Tentative name of Tab Images. This is where the user will be able to manage the Organizations Images. Functionality will be defined in the UI section.

# UI

New Organization Image Tab

A screenshot of a computer

Description automatically generated

From the details tab, we will remove the missing image field.

The tab will populate with each available image category (Missing Image, Builder Logo). Under the category there will be:

* the image assigned to the category with buttons to update or delete
* no image and the button to Upload an image to that category.

Clicking Upload will show a file explorer to allow the user to select an image file to upload. We are limiting the image file type to JPG, JPEG or PNG. On successful upload, the category will show the image and display the Upload and Delete buttons. When a user tries to upload a file that is not supported, the app will show an error dialog stating that the file is an unsupported type. When a user tries to upload a file that is too big, the app will show an error dialog stating that the file is too big.

Clicking Update will, in a similar fashion to the Upload button, allow the user to upload a different image to the category. The same errors that constrains the upload will be adhered to in the Update.

Clicking Delete will delete the image record. For the category, it will then hide the Update and Delete buttons and show the Upload button.

# QA

Points to test:

* VEO Admin: New Organization Image tab
  + Loading images correctly to the page.
  + Uploading an image successfully as well as:
    - App errors on file type
    - App errors when file is too big
  + Updating an image successfully.
    - Should be able to change image as well as image type.
    - App adheres to the same error as Uploading
  + Deleting an image removes the image from the database as well as screen.
* Confirmation that the images usage throughout the application is correct.
  + Confirmation that the app is displaying the correct Missing Image
    - Cabinet Selection Wizard
    - Design Session Wizard
    - Dmh
    - Product catalog
    - Option Pricing
  + Logos
    - Application Header
    - Reports

# Other Topics

When limiting on size, do we want to be able to compress it if it is too big?