

Niagara Falls CSD Conference Requests Documentation

A PowerApps app with a PowerAutomate flow for conference requests & management

# Overview

* What is the app and what does it do?
  + *This app is a customized SharePoint form built with Microsoft PowerApps. The form fields were created using the existing request form and columns were added to make the app more functional and to create the PowerAutomate Flow that is associated with the app.*
* Who are the users and what are their roles?
  + *The main users of the app will be faculty and staff of the Niagara Falls City School district. Users will be broken up into two distinct roles; User and Admin. Admin have the ability to edit requests, approve requests, deny requests, and delete requests. Users can only create forms/requests (see future improvements for more information)*
* What are the benefits and objectives of the app?
  + *The main benefit of the app is that it allows users to upload conference documentation, generate automatic approvals and provide a custom experience based on the needs of the district. The objective is to streamline the request process and reduce the amount of paperwork required while still maintaining a database of requests for auditing and record keeping.*

# Features and Functions

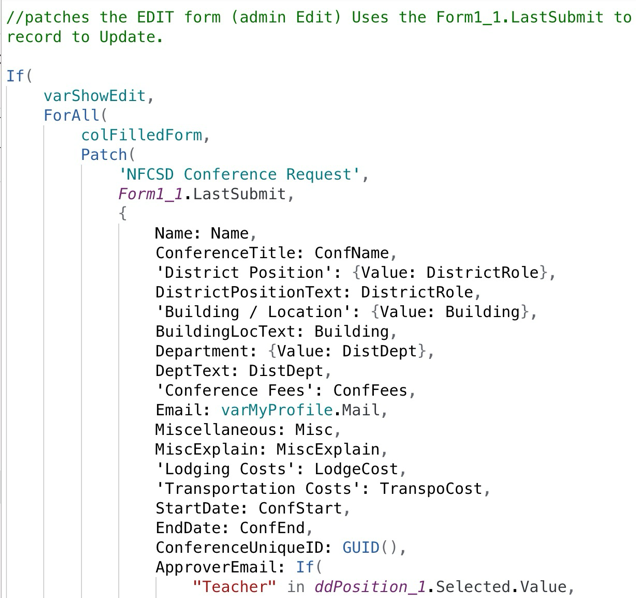
* How to access and launch the app?
  + The app will be accessed via direct link. On first launch, users will be required to select allow to the connected services. This includes SharePoint, Outlook, and Office365.
  + The app will need to be shared with users using the PowerApps share interface.
    - Link: [Niagara Falls CSD Conference Requests](https://apps.powerapps.com/play/e/default-5a228b8d-6047-43a7-8aa3-1517a2972f84/a/781f7164-137a-4c23-b2b7-0c3cc543c3b0?tenantId=5a228b8d-6047-43a7-8aa3-1517a2972f84&hint=89424980-03cc-4982-897c-c0fbf7a5424e&sourcetime=1711210835182)
* How to navigate the app interface?
  + *The app interface includes clearly labeled and defined buttons. Back/forward arrows, review, submit, edit, and more.*
* How to create, edit, delete, and submit conference requests?
  + *To create a new request, users will select the labeled button on the left side of the main screen.* 
    - *The user will be prompted whether they are filling it out for themselves or someone else. If someone else is chosen, a search bar to find and select a user will appear.*
  + *From there, the app will navigate to the form screen that includes a + button to begin their request.*
  + *Fields that are required will be highlighted in red and the review button will be disabled until all required fields contain information.*
  + *In its current state, the form is meant to be completed in a single session. The ability to save locally or on the server can be added over time.*
  + *On the review screen, users will see a document that contains all the information they entered to verify accuracy. They are also prompted to verify they have attached any required documents. Plans include the ability to print this document.*
  + *Currently, users cannot edit submissions. That feature will be added over time.*
* How to view, approve, reject, and comment on conference requests?
  + *To view/approve/reject, the app takes advantage of Microsoft Approvals via Microsoft PowerAutomate. The approval structure is based on the requested process. For teachers, it goes to their building principal first. For all other staff, it will go directly to their department head. This information is stored in two separate databases-one that has the building and principal and another that has the department heads. Each approval is sent based on the response of the previous approver. If denied, an email is generated to the requestor saying the request was denied. The approval has 4 or 5 steps depending on the person requesting.*
* How to generate and export reports on conference requests?
  + *Exports can be generated directly from SharePoint. Over time, reports will be able to be generated directly in the application. In SharePoint, the requests can be exported by selecting the export button.*
* How to troubleshoot common issues and errors?
  + *Troubleshooting errors will be handled by DD*
* Future Improvements?
  + There are several improvements planned to the application. These improvements will all be designed with ease of use and increased efficiency in mind. \*\*Some of these improvements have been mentioned in other areas of this document. \*\*
    - Save before submission: This improvement will let users save their progress on a form and return later to submit the app for approval. This will be handled with a simple number column (0 or 1). When selecting save, the state will be set to 1. When submitting, the state will be set to 0. The flow must be changed to check this column's condition before running the approvals. When submitting a saved form, it will delete the original and create a new form. This will allow the current trigger to still work and reduce the amount of modification needed.
    - User viewing/editing: this improvement will allow users to go in and view their form and make modifications to their submission if required. This will require the addition of a new screen and some variables.
    - Printing: this improvement will allow users or admin to print a completed form to keep for their records. As an alternative to printing, the file could be saved as a PDF.
    - E-Signatures: If desired, a place to enter a digital signature can be included in the application.

# Components and Dependencies

* What are the data sources and connections used by the app?
  + *The data sources are three SharePoint lists stored in the conference requests SharePoint site. The main SharePoint site will need to be shared with users via SharePoint’s share menu. The databases have the following permissions: users can only create and view items they created. Admin can edit and view all items. For the databases that contain the buildings, principals, and department approvers, they can only be viewed and edited by admin on the site. Having the buildings/Principals and department approvers as separate lists, it allows changes to be made without the need for updating the app code.*
  + *Supporting documents: There is one supporting document that will be linked in the app. This document contains the funding codes for conferences that require a substitute.*
  + *Links:*
    - *Requests:* [*https://nfschools.sharepoint.com/sites/ConferenceRequests/Lists/NFCSD%20Conference%20Request/AllItems.aspx*](https://nfschools.sharepoint.com/sites/ConferenceRequests/Lists/NFCSD%20Conference%20Request/AllItems.aspx)
    - *District locations w/ principals:* [*https://nfschools.sharepoint.com/sites/ConferenceRequests/Lists/DistrictLocations/AllItems.aspx?viewid=858808e9%2D26bf%2D4d1e%2D917d%2Dd6e70207d181*](https://nfschools.sharepoint.com/sites/ConferenceRequests/Lists/DistrictLocations/AllItems.aspx?viewid=858808e9%2D26bf%2D4d1e%2D917d%2Dd6e70207d181)
    - *Department Approvers:* [*https://nfschools.sharepoint.com/sites/ConferenceRequests/Lists/DeptApproverEmails/AllItems.aspx*](https://nfschools.sharepoint.com/sites/ConferenceRequests/Lists/DeptApproverEmails/AllItems.aspx)
* What are the screens, controls, and formulas used by the app?
  + *The app consists of the following screens: ￼*
    - *Welcome screen*
    - *Request form screen*
    - *Review/Submit screen*
    - *Edit screen (admin only).*
  + *The app uses the following functions:*
    - *Set() for variables*
    - *Text() for formatting*
    - *HTML() for generating a bulleted list*
    - *Sum() for generating estimated totals*
    - *Patch({}) for transmitting data*
    - *ClearCollect() to store local versions of datasources*
    - *Collect() to store temporary conference requests for review.*
    - *SubmitForm() to submit the attachments*
    - *If() for logic trees*
    - *ForAll() for looping through collections and patching the matching columns.*
    - *GUID() for generating unique IDs for conference requests.*
* What are the variables, collections, and contexts used by the app?
  + ***Variables***:
    - *varStep: Integer that stores the current step of the form. This is used to make the correct portion of the form visible. It ranges from 0-3.*
    - *varMyProfile: stores the user’s profile at launch. This reduces the number of calls to the Office365 API.*
    - *varItemToEdit: this variable stores a record of the current form that is being edited.*
    - *varNewForm: Boolean variable for the app to know if a new form is being used.*
    - *varReviewForm: Boolean variable for the correct controls to appear on the review screen.*
    - *varShowOtherPerson: Boolean variable that is used for determining who is filling out the form and using the correct name and email for the request.*
    - *varOtherPersonName: Record variable that stores the profile information of a selected user when creating form on behalf of someone else.*
    - *varTeacherApproverEmail: text variable used to get the email address of the correct principal based on the building selected.*
    - *varShowEdit: Boolean used to determine if the edit window should appear.*
    - *varShowPersonSearch: Boolean, unused.*
    - *varShowRemoveWarn: Boolean to have the remove warning appear when delete button is pressed.*
    - *varShowCancelWarn: Boolean used to have the text change to remove user of cancellation a form.*
  + ***Collections***: Collections are tables that can be accessed directly in the application memory, speeding up access times.
    - colBuidlingPrincipals: *collection containing a local version of the SharePoint list. This limits the API calls during a session. Collection is updated during app launch to receive the most up-to-date data.*
    - *colFilledForm: collection containing all of the information a user entered for review. Used to patch database when form is submitted.*
    - *colDeptEmails: contains a table that has each department approver with an id and department name. This collection is generated at launch from the SharePoint list containing this information.*
* What is the PowerAutomate flow and how does it work?
  + *The flow is called NFConfApprovals. The flow works by being triggered when a new item is created on the Conference Approvals list. A 25 second delay is added to ensure all information reaches the list before the ‘Get item’ function is used to grab all information from the submission.* 
    - *The flow then checks to see if the user submitting is a teacher.* 
      * *If yes, first approval sent to building principal*
      * *If now, first approval sent to department head*
    - *The flow sends approvals to designated people in HR, administration, deptartments, etc. This is done only after the previous flow is approved using a conditional ‘if’ in the flow where the condition is ‘Outcome’ (of previous approval) is “Approve”*
    - *An additional document containing a visual breakdown will be included.*
* What are the security and compliance considerations for the app?
  + *For security and compliance, the app uses the district’s Office 365 tenant. This means the security already used by the district is implemented into the application. It is internal facing only and cannot be accessed by users without a NFCSD domain account and license.*
  + *For data governance, the app governs the databases based on user account level. This will prevent basic users from seeing, deleting, or editing other users' form.* 
    - ***Important note: The list will always be directly accessible to users without going through the app. This can be mitigated by using a PowerShell command to effectively hide the direct list from users.*** 
      * ***Generally, with the data governance already in place and with user training, it will not be an issue.***
      * ***If further governance is required, the purchasing/subscribing to Microsoft Dataverse would be required.***

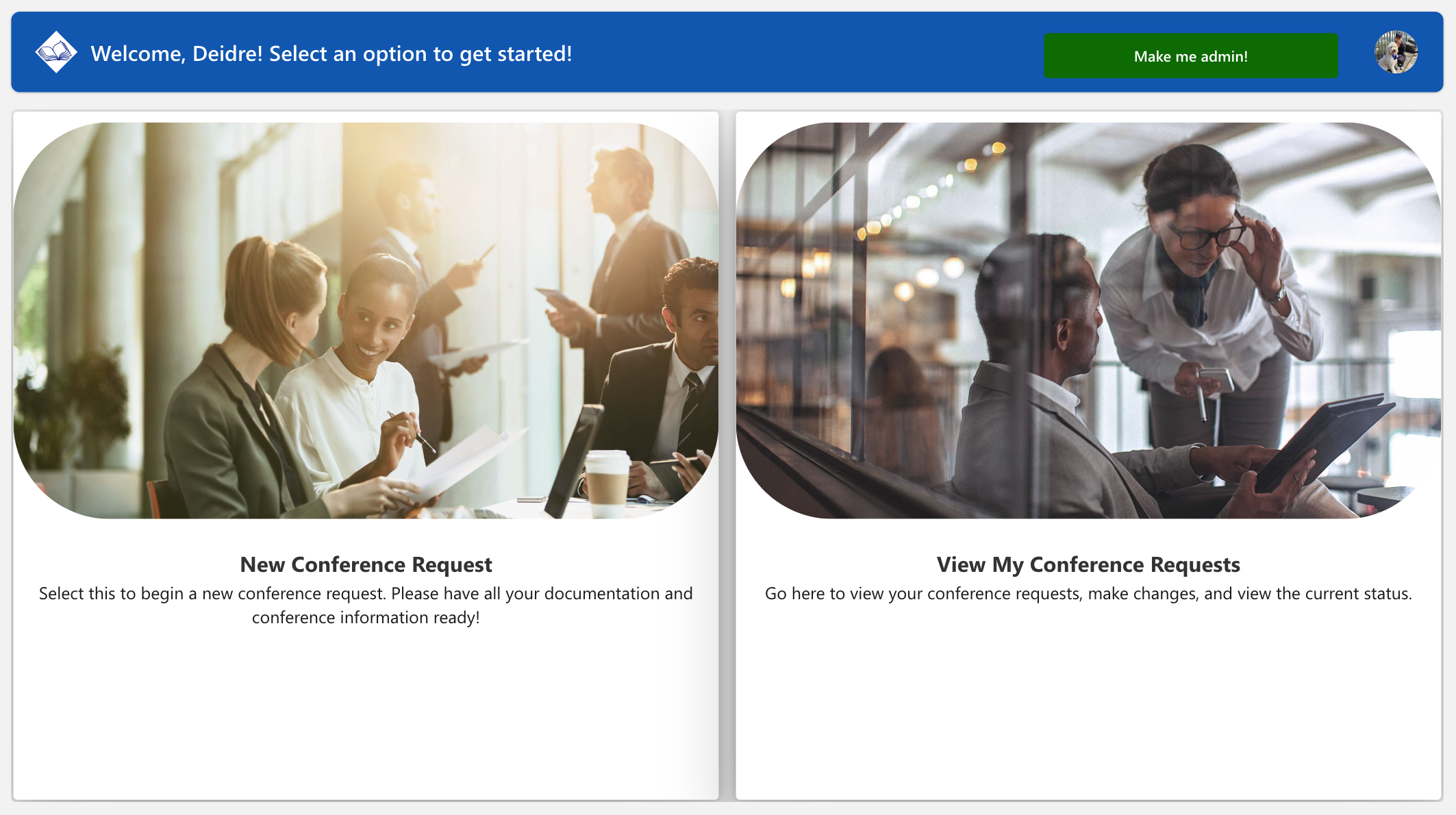
Code Snippets:

A screenshot of a computer program

Description automatically generated

Screen Captures:

Teacher View:



Admin View:

