Legend

Mandatory field or process	
Triggered after a specific notification event	
Non-mandatory field	

10-22-2022

Feature Workflow

This feature workflow example is from my own personal future product.

It is not complete and will require alterations.

Displayed:
- An event creation process

Included:
- Workflow
- API calls
- Display/view information
- Table names
- SQL instructions
- Notification Instructions

Created In:
- Miro

Flowchart

3 View/Create a new event

- View a saved event → **3A View existing event**
- Create a new event → **3B Create/update an event**

3A View existing event

Logged events can be filtered by
- Newest added
- Oldest added
- Event Type
- Event Theme
- Event Start Date (1st)
- Application Submitted Date
- Application Approved Date

Display events per filter/sort
- View event(s)
- Update event(s) w/warning
- View attached list(s)
- Update attached list(s) w/warning
- Delete attached list(s) w/warning

SAVE | UPDATE

(clarity in business logic section)

→ **2 User Dashboard**

3B Create/update an event → **[FORM]**

Event Name
- Purpose: Naming of an event
- Put into **Events** Table,
- SQL: **event_name** VARCHAR(50) NOT NULL
- System: Display prompt box
- Field validation: AlphaNumeric - + !

Event Type
(Will be a checklist)
- Purpose: Allow user to choose what event type this event falls into
- Get from **Type** Table
- Put into **Events_Type** Table
- System: Display **Types** Table via a checklist.
- Field validation: User **must choose at least one**, but can choose as many as they wish

Event Theme
(Will be a checklist)
- Purpose: Allow user to choose what event theme this event falls into
- Get from **Theme** Table
- Put into **Business_Theme** Table
- System: Display **Theme** Table via a checklist.
- Field validation: User **must choose at least three**, but can choose as many as they wish

Enter in event address
- Purpose: Store location of event space.
  - Future use's
    - Can include displaying an event within a set distance parameter for another vendor to find the event.
    - events can reach out to vendors to own their own vendor pages which will auto-sync with vendors who have a already added them to their event calendars and adjust pre-defined user notifications.
- Put into **Venue** Table (look to ERD for specifics)
- System: Display address entry form. See field examples below.
- Field validation: **City, State, and Zip are mandatory fields**

Venue Name	Venue Website	Venue Reservations Phone Number	Venue Reservations email address
Address Line One	Address Line Two	City	State/Province	Zip Code

- Note all appropriate validations
  - website
  - phone number
  - email address
  - **Should we add country?**
  - dropdowns for city/state/prov

Event Website Address
- Purpose: Allow user to save website address for review
- Put into **Events** Table,
- SQL: **event_www** VARCHAR(75) NOT NULL
- System: Display prompt box
- Field validation: https://..........

Event Instagram
- Purpose: Allow user to save IG
- Put into **Events** Table,
- SQL: **event_ig** VARCHAR(75)
- System: Display prompt box
- Field validation: https://.......... or @XXXXXX

Event Facebook Link
- Purpose: Allow user to save FB
- Put into **Events** Table,
- SQL: **event_fb** VARCHAR(75)
- System: Display prompt box
- Field validation: https://..........

Event Contact Name Name
- Purpose: Allow user to save contact person
- Put into **Events** Table,
- SQL: **event_contact** VARCHAR(100)
- System: Display prompt box
- Field validation: Alpha only, allow spaces and hyphen

Main Contact Email Address
- Purpose: Allow user to save contact direct email address
- Put into **Events** Table,
- SQL: **event_ctemal** VARCHAR(100)
- System: Display prompt box
- Field validation: email address validation: @xxx.xxx or @xxx.xx

Main Contact Phone Number
- Purpose: Allow user to save contact direct email address
- Put into **Events** Table,
- SQL: **event_phone** VARCHAR(15)
- System: Display prompt box
- Field validation: Numeric only - phone validation for US/UK formats

What are the dates of the show?
- Purpose: Store start and end dates of event
- Put into **Events** Table, , (event_strtdate2, event_enddate2 etc)
- SQL: **event_strtdate1** DATE
- SQL: **event_enddate1** DATE
- System: Display entry field form. See field examples below.
- Field validation: **Need some indicator to add up to 6 unique start/stop dates because some events are multiple weekends. Think Renaissance Faires!**

Start Date 1	End Date 1	→ System: Pick a date option please
+ add more		

Is there an application process?
No there isn't an application process?
- Purpose: Some events have an application process. The purpose here is for the user to save when an application process opens and closes.
- Put into **Events** Table,
- SQL: **event_appyn** TINYINT(1) DEFAULT 1 NOT NULL
- System: Display entry field form. See field examples below.
- Field validation: **User must choose yes or no!** If checked yes, go to date fields. If no, move forward.

Y/N — Checked Yes →

Start Date 1	End Date 1

- Put into **Events** Table,
  - SQL: **appstart** DATE
- Put into **Events** Table,
  - SQL: **append** DATE
- System: Pick a date option please

What are the available table or booth configurations available at the show?
(Will be a checklist)
- Purpose: Allow user to record what types of booths / spaces are available at this event. May include pricing.
- Get from **BoothSpace** Table
- Put into **Events_BoothSpace** Table
- System: Display **BoothSpace** via a checklist.
- Field validation: User **must choose at least one**, but can choose as many as they wish

Do you have existing lists to add to this event?
Do you want to create a list to add to this event?
Do you wish to add lists later to this event?
- Purpose: Allow user to choose from
  - a - Existing saved lists
  - b - Create a new list
  - c - Bypass this until they are accepted into the event
- Table: TBD
- System: Display Event Space via a checklist.
- Field validation: User **must choose at least one selection if they chose a or b**, but can choose as many as they wish. Yes, they can add existing lists and create a new list.

Y - have list — Yes list →
- If user wishes to choose an existing list go to 4A
  - Display active lists list option 4A with check marks
- If user wishes to create a new list go to 4

Not right now

Is this an annual event?
No this is not annual event?
- Purpose: By the user stating "yes" this is an annual event, backend processes and logic creates future notifications
- Put into **Events** Table,
- SQL: **event_annual** TINYINT(1) DEFAULT 1 NOT NULL
- System: Display just these questions with the check boxes
- Field validation: User **must choose at only one selection. If they chose no a warning will pop up and can be X'd out.**

Y/N → Notification process

Not an annual event

Load In Date and Time
- Purpose: Upon acceptance to an event, load in / out information is normally shared with the vendor/user
- Put into **Events** Table,
- SQL: **event_lidate** DATE
- SQL: **event_litime** TIME
- SQL: **event_lodate** DATE
- SQL: **event_lotime** TIME
- System: Display entry field form. See field examples below.
- Field validation: Load In Date & Time & Load Out Date & Time

Load In Date	Load In Start Time
Load In Date	Load Out Start Time

Notes:
- Purpose: This is a place for the user to make notes about the event.
- Put into **Events** Table,
- SQL: **event_note** TEXT(1000)
- System: Display entry field form.
- Field validation: None

SUBMIT BUTTON