QuestBoard Ver. 2 Roadmap

Objective
Manage club operations within a website.

Context
Tabletop Adventuring Club (TAC) is a West Marches-style Dungeons & Dragons (D&D) gaming club. Dungeon Masters post their games on Meetup, and Players sign up for games on Meetup. Most operations are run in Discord. Players post their characters, post questions, or look up club or lore information on the Discord server. The club has rules for building characters for games and character advancement. The club has a process for becoming a DM. They have their own factions, magic items, and prices for magic items. DMs must log the results of their games. DMs are required to run a game once a month in order to remain Active.

Workflow & Database Structure

Developing the workflow and database structure is the crucial first step before you even think about the web app's structure or coding. This foundational work will dictate how smoothly and effectively your application functions.

Step 1: Develop the Workflow

This involves mapping out the actions and interactions of your different user roles. Think of it as telling a story about how each person uses the system.

User Roles

Each subsequent role has all the rights of previous roles.

- **Guest:**
  - View public information (game calendar, DM list, character progression process, factions)
- **Player:**
  - Create/edit their character profiles.
  - View their characters' progression (level, items, faction status, etc.).
  - View & sign up for available games (calendar/form).
  - View DM list.
- **Initiate Dungeon Master (Initiate):**
  - Create games they run (date, title, description, system, location, level range, DM oversight, faction credit)

    - Log results of games they run (Brief Summary, Notable Plot Points, Loose Ends, Rewards given, Notable NPC details, Detailed Notes (optional)).
    - Create/manage their DM profile.
    - Initiates cannot create Factions, but they can award Faction credit for existing Factions
- **Dungeon Master (DM):**
    - Create games they run (date, title, description, system, location, level range, DM oversight, faction credit)
    - Log results of games they run (Brief Summary, Notable Plot Points, Loose Ends, Rewards given, Notable NPC details, Detailed Notes (optional)).
    - Create/manage their DM profile.
    - Create and manage their factions.
- **Moderator (Mod):**
    - Higher level authorities but subordinate to MP authorities (can do almost anything an MP can - the main distinction is MPs are paid owners and Mods are volunteers, usually with specific roles such as "Creative Director")
- **Managing Partner (MP) (~ admin):**
    - View all game logs.
    - View all player character profiles.
    - View statistics on games, players, and characters.
    - Manage user accounts.
    - Manage DM progression.
    - Edit any data as needed (e.g., correcting DM errors).
- **Administrator**:
    - Complete access for the administrative operation of the technical functions of the application.

Workflow Overview Table

Workflow	Trigger	Inputs	Outputs	Stakeholders	Notes / Questions
New DM Onboarding	Mod+ clicks “DM Onboarding”; Prereqs: Player status (# games?), “DM Orientation” attendance	Date Player attended DM Orientation. Sends link for Player to edit their DM Profile. DM Profile: Pulls info from Player record (Name, Discord Handle, email/auth); Player enters Full name, Preferred name, Meetup name, system(s), max table size (4-8)	DM record in “Initiate” status, email confirming; notification to Mod+	Initiate, DM, Mod, MP (link only available to Mod+)	Mod/MP can override prerequisites, so prerequisites should be displayed (a warning notice if not met); otherwise, no checks to prevent onboarding except must be done by a Mod+
Profile Update	Player+ clicks “User Profile”	Fields requiring update (Player required fields: Full Name, Discord Handle; (Initiate+ also require: Meetup Name, System(s), Max Table Size)	Updated User Profile (only Mod+ can edit “DM Status”)	All non-guests (Players/Initiates /DMs only edit their profile, Mod+ can edit all)	MP/Mod uses for Advancement from Initiate to DM & to correct errors
New Player Onboarding	Player clicks “Register” ~~or submits first character~~	Full Name, Preferred Name (optional), Email, Discord Handle	Player record, welcome email with profile link	Player, MP, Mod (anyone can register)	Will you verify emails? Yes, if not using Google Auth Let them edit their own profile? Yes, except for some fields (only Mod or MP can update “DM Status”)

Workflow	Trigger	Inputs	Outputs	Stakeholders	Notes / Questions
DM Game Creation	DM clicks "Add Game"	Date, title, description, system, location, level-range, oversight DM, faction credit	New Game record, calendar slot, "Game" choice in Player form	DM, Players (Initiate+ can access)	What prevents date collisions? (system checks - see workflow) Do you enforce a max-table-size? No - this is managed in Meetup (maybe we don't need to track this in the app, but we'll keep it for now)
DM Game Editing	DM clicks "Edit Game" → dropdown of the DM's games in descending order	Date, title, description, system, location, level-range, oversight DM, faction credit	Updated Game record, update "Game" choice in Player form (if necessary)	DM, Players (DM+ can edit any, Initiate can only edit their own)	DMs can edit details of their game, including deleting it. If they edit something that changes the "Game" choice, need to update that.
DM Game Results Logging	After the game ends, DM clicks "Log Results"	Brief Summary, Notable Plot Points, Loose Ends, NPCs, rewards, detailed notes, faction advancement approvals, confirm character (remove those who didn't attend)	Game log entry, Process character advancement & DM Variety Benefit, Send notifications on character status	DM, Players (Players & Guest only see recap portion - Brief Summary, Notable Plot Points)	Do players get notified? No Will logs ever be editable or locked after X days? Editable.
DM Game Results Editing	DM clicks "Edit Results" from list	Edit fields (listed above)	Updated Game log entry, if applicable: update character advancement & DM variety benefit & send applicable notifications	DM, Players (same as above)	

Workflow	Trigger	Inputs	Outputs	Stakeholders	Notes / Questions
DM Faction Creation & Editing	DM clicks “Faction Management”	Faction name, Faction leader, description, membership rules, level-up requirements & benefits	Faction record, open for character petitions	DM, Players (DM+ can access)	Do factions have leaders? Yes (NPCs) Membership limits? No
DM Faction Management	DM clicks “Faction Management”	Requests for faction assignment by Players	Updated Faction assignments	DM, Players (DM+ can access)	
Player Character Creation & Editing	DM or Player clicks “Character Management”	Character stats and equipment: Name, Build, Level, Species, Class, Subclass, HP, AC, PP, STR, DEX, CON, INT, WIS, CHA, Magic Items	Character record	DM, Players (Guests cannot access)	Are there any approvals required? I don’t think so. DMs can edit characters to correct issues, if necessary
Player Game Entry	Player clicks “Upcoming Games”	Select game, select/update/create character, Drop-in game (yes/no)	Updated Game record	DM, Players (Guests cannot access)	
Game Report	DM/Player clicks “Game Report”	Pulls from Game data	Game Report: list of characters in the game (with character stats & player name), average character level, character level range, and (in the future maybe) experience range for Easy, Medium, Hard, and Deadly encounters	Everyone (Player Name hidden from Guest view)	

Workflow	Trigger	Inputs	Outputs	Stakeholders	Notes / Questions
Character Faction Request	Player clicks "Join a Faction" (only available after prerequisites met)	Select factions available	Faction request notification to owners of selected factions	Players, DM Faction Owners (Faction Owners, MPs, or Mods can approve)	Is there any need for Players to edit their faction information? I don't think so.
Player Character Progress Report	Player clicks "Character Progress Report", then selects their character (if more than one)	Pulls from Character & faction data	Character Progress Report, including # games/level played, # games/level needed to advance, Faction status, and DM Variety Benefit status	Players (Players can only see their characters, DMs and above can see all characters)	3 columns for each report (character, faction, DM variety benefit)
DM Games Dashboard	DM clicks "DM Games Dashboard"	Pulls from Game & Character data	DM Progress Report: - # Games each month for previous 12 months - "Journey" status (progress from Initiate → DM)	MP, DMs (Players and Guests cannot access)	Used primarily by MPs to manage club DM requirements, but DMs have access to see their own progress
Pending Requests Report	DM+ clicks "Pending Requests Report"	Pulls from all pending requests	Pending Requests Report: date of request, requester, requested thing, designated approver(s)	All non-guests	Used primarily by MPs and Mods to ensure requests are being processed
Admin Meeting Creation & Editing	MP/Mod clicks "Create/Edit Meeting"	Select date, meeting type, host, location	Meeting in calendar	MP/Mod, DMs (only MPs & Mods can create/edit)	Monthly DM Council Meetings, DM Orientation meetings, Lore Meetings, etc.

Workflow	Trigger	Inputs	Outputs	Stakeholders	Notes / Questions
Password Reset	DM/Player clicks “Reset Password” on login screen	DM/Player’s email → checks for email in Players data	Either “no email registered, please register” or “Password reset email sent, check email” → DM/Player sent email with link to reset password (removes old password)	All non-guests	Add to a log entry within the player profile each time this happens.

Issues:

- **Do I have an existing Auth system?** No. I want to use Google~~ primarily, but allow non-Google users to use email/password~~.
- **Notifications (SMS, Discord, Meetup, integrations)**: I want to add integrations with Discord and Meetup later. I’ll think about SMS.
- **Approval gates**: Most things will be self-service, but I want to keep all records so we can see all changes. This means that every change to a character will create a new entry, and the old entries will become historical but not be deleted. The editor should be logged in, too.
- **Data ownership**: Unless otherwise indicated, most things can be edited by anyone of the same authority level.

Authority levels:

1) **MP**: Managing Partner (full administrative access, edit rights to everything)
2) **Mod**: Moderator
3) **DM**: Dungeon Master
4) **Initiate**: DM-with-oversight
5) **Player**
6) **Guest** (view only to some reports & general club information)

Webpage Hierarchy

Menu [sidebar] (first 2 levels are menu items, 3rd level is the screens list)

- Games
    - Upcoming Games (Join a Game!) [Visible to: Player+]
        - Player Game Hub screen
        - Game screen with Game Report screen embedded (for Join a Game process)
        - Game screen with Game Report screen embedded (above) and Character screen embedded (below)
    - Game Management [Visible to: Initiate+]
        - Game Management screen
        - Game screen
    - Game Report [Visible to: all Users]
        - Game Report screen
- Characters
    - Character Management [Visible to: Player+]
        - Character Management screen
        - Character screen
    - Character Progress Report [Visible to: Player+]
        - Character Progress Report screen
- Factions
    - Faction Management [Visible to: Player+]
        - Faction Management screen
        - Faction screen
        - Faction Membership screen (Faction screen embedded above)
    - Join a Faction [Visible to: Player+]
        - Join a Faction screen
        - Faction screen (similar to Faction Management/Faction screen)
- Administration
    - User Profile (edit Player/DM Profile) [Visible to: Player+]
        - User Profile screen
    - Register [Visible to: all Users]
        - User Profile screen (same as above, but blank initially)
    - DM Onboarding [Visible to: Mod+]
        - New DM screen
    - DM Games Dashboard [Visible to: DM+]
        - DM Games Dashboard
    - Meeting Management [Visible to: Mod+]
        - Meeting Management screen
        - Meeting screen
    - Pending Requests Report [Visible to: Initiate+] (Initiate & DM only have access to their pending requests, Mod+ have access to all pending requests)
        - Pending Requests Report screen

… “Login” button… top right

End-to-end flow

State Machine Concepts:

- **State**: condition of an object
- **Event (Trigger)**: an occurrence that affects the State
- **Event Parameter**: attribute of the event
- **Guard Condition**: Boolean conditions attached to transitions. If multiple transitions share the same event, guards determine which path is taken.
- **Transition**: relationship between two States, depicted by an arrow, labeled by the event (trigger). A transition fires when the associated event occurs and no guard conditions are met.
- **Action**: instant behaviors executed as part of transitions or on entering/exiting a state, e.g. sending a confirmation email or assigning a character to a slot.

Screen displayed
Table pulled from or edited

New Player Onboarding

Anyone can become a player by just creating a User (Player) profile.

Until all required fields are properly completed, the system won't save the new User Profile.

Google Authentication is a part of this process. New Users will have to 1) have a Google account and 2) allow Google authentication for this application.

See the Update Profile workflow for a more complete list of DM Roles, DM Statuses, and Profile Statuses.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Guest	Menu: click "Register" under "Administration"	n/a	Show empty "User Profile" screen with "Save" and "Cancel" buttons
2a	Guest	Click "Cancel"	n/a	Show main screen (no User created)
2b	Guest	Fill in fields & click "Save"; If any required fields are blank, show pop-up "Please complete all required fields" with "Ok" button	Add User Profile record; 🔁Status = Complete 🎮DM Role = Player	Show main screen

State: N/A (Guest Users have no User Profile, so the State is non-existent)

Event: Guest clicks “Register”
Event: Guest fills in the required fields
Event: Guest clicks “Cancel” → return to main screen, OR
Event: Guest clicks “Save” →

Guard Condition: Ensure all required fields are completed properly; ensure User approved Google authenticate; If either is not complete → display pop-up warning, return to edit screen. If both are completed →

State: User Profile created, Status: Complete, DM Role: Player → go to main screen; display DM Role with User info on QuestBoard

New DM Onboarding

Only Mods and MPs will have permissions to onboard new DMs.

If the Mod/MP has all the player’s profile information (perhaps because they collected it during the Orientation meeting), then they can enter it themselves, completing the DM Profile. If they do not have all the information, they can get the DM Profile started, then save it as a Draft. The system will then email the Player to ask them to complete their profile.

Mod+ can change Players to Initiates and Initiates to DMs at the appropriate time. They can also skip (override) the normal process when necessary (at Step 2 below)

🎮DM Roles: Player, Initiate, DM, Mod, MP (a Mod+ determines these)

👷DM Statuses: Active, Inactive, LOA, Pending, Banned (all Initiates are in Pending status; LOA = “Leave of Absence”, which is an approved break from DMing - an alternative to being Inactive)

Profile 🔁Statuses: Complete, Awaiting User (these are system-controlled but viewable to User)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Mod+	Menu: click “DM Onboarding” under “Administration”	Read “Player” User profiles	Show “New DM” screen with Prerequisites and editable fields & Player List dropdown as a table for *entering multiple new DMs at once* (Mod+ can override Prereqs, so they are displayed for informational purposes only)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
2	Mod+	Fill in fields known & click “Save” If any required fields are missing, show a pop-up to confirm the Mod+ wants to proceed (this means the Mod+ doesn’t have all the information, and the Player needs to complete their profile)	Update User profiles with DM fields (🎮DM Role = “Initiate” or “DM”, as appropriate) 👷DM Status = Pending (🔁Status = Complete if all fields complete, or 🔁Status = “Awaiting User” if any required fields are incomplete)	If any required fields are blank (Full Name, Meetup Name, System(s), Max Table Size), send email to Player with link to update their DM profile; If profile complete, send confirmation email to Player
3	Initiate or DM [if profile not complete]	Click “Complete DM profile” link in email OR Menu: click “User Profile” under “Administration”	Read the User profile	Show User profile with editable DM fields (required fields are indicated as such)
4	Initiate or DM [if profile not complete]	Edit fields, fill in blanks, click “Save”	Update User profile with DM fields (🔁Status = Complete if all fields complete)	Show main screen; Send confirmation email to Player

Update Profile

This workflow is used by all Players+ to manage their own profiles and also for Mod+ to manage other user roles.

Only MPs can designate a Mod.

Admin is the system administrator with full rights.

🎮DM Roles: Player, Initiate, DM, Mod, MP (a Mod+ determines these)

👷DM Statuses: Active, Inactive, LOA, Pending, Banned (all Initiates are in Pending status; LOA = “Leave of Absence”, which is an approved break from DMing - an alternative to being Inactive)

Profile 🔁Statuses: Complete, Awaiting User (these are system-controlled but viewable to User)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Player+	Menu: click “User Profile” under “Administration”	Read User profile	Show “User Profile” screen
2	Player+	Edit fields, click “Save”; show pop-up “Send email verification?” with “Yes” and “No” buttons	Update User Profile	Show update “User Profile” screen; if “Yes” to pop-up: send confirmation email

DM Role State Machine

Each user can be in one of the DM roles:

- Player → Initiate → DM → Mod → MP

Transitions:

- `Create Profile`: Player → Initiate (manual by MP/Mod)
- `Promote`: Initiate → DM → Mod → MP (all manual by Mod+ or MP)
- `Demote`: Any state → any lower state (also manual)
- There is no automatic progression.

```
[Player]
  └──(Mod+ promotes)──▸ [Initiate]
      └──(Mod+ promotes)──▸ [DM]
          └──(MP promotes)──▸ [Mod]
              └──(MP promotes)──▸ [MP]
```

DM Status State Machine

Status is tracked independently of DM role:

- Pending, Active, Inactive, LOA, Banned

Transitions:

- `Monthly Game Logged`: Inactive → Pending (for Initiates) or Active (for DM+) (auto)
- `1 Month of No Games`: Pending or Active → Inactive (auto)
- `Manual Approval`: Inactive, Pending, or Active → LOA (Mod+)
- `Manual Ban`: Any state → Banned (Mod+)

[Inactive] ──(Logs game)──▸ [Pending (Initiate) or Active (DM+)]

[Pending (Initiate) or Active (DM+)] ──(No games 30d)──▸ [Inactive]

All states ──(Mod bans)──▸ [Banned]

All states ──(Mod sets LOA)──▸ [LOA]

Profile Status State Machine

States: Awaiting User → Complete, Suspended, Banned, Inactive
The system only allows a Complete status if all required fields are entered.
If the User attempts to delete required fields and save their profile, the system will reject the request and do nothing (with a pop-up notice).

Transitions:

- `User completes required fields`: Awaiting → Complete (system)
- `Admin sets up profile for user`: system creates profile in Awaiting state (unless all required fields are complete, then it will be created in Complete state)
- `User deletes required field`: System does not allow updates to be saved
- `Admin adjusts state`: Admin/Mod+ can move a Profile status to Suspended, Banned, or Inactive manually, when necessary

[Awaiting User]

  └──(Fields completed)──▸ [Complete]

[Complete] ──(Admin manually adjusts)──▸ [Suspended, Banned, or Inactive]

Game Creation, Editing, and Results Logging Workflow

🔁Status options: Scheduled Draft, Scheduled, Logged Draft, and Logged.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Initiate+	Menu: click “Game Management” under “Games”	Read User profile	Show “Game Management” screen with Game list dropdown and “New Game” button
2a	Initiate+	“Game Management”: click “New Game”	Write new Game record (🔁Status = “Scheduled Draft”, Version = 0, “Last Modified” = Now, “Modified by” = User)	Show “Game” screen with blank data fields

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
2b	Initiate+	Select Game from dropdown (optional: use filters to find just User's Games)	Load game data	Show "Game" screen with data in fields
3	UI Update	If 🔁Status = "Scheduled Draft" or "Scheduled", show buttons: "Save Game to Post" and "Save Draft Game to Post"; If 🔁Status = "Scheduled", "Logged Draft" or "Logged", show buttons: "Save Results Log" and "Save Draft Results Log"	← UI update Also, show the Calendar view below & adjust for the date selected (show a week always starting with Monday, initialize to the current day, then adjust the week displayed when the User selects a Game Date) (this allows the User to see if the date is already filled)	← UI update
4a	Initiate+, System Check	Click "Save Game to Post" or "Save Draft Game to Post"	If 🔁Status = "Scheduled Draft" or "Scheduled": Check for date collisions (4 games allowed per location per day); Else: skip	If ≤ 3 games on selected date and location, proceed to step 4; else, display pop-up "There are already 4 games scheduled for this location on your selected date. Schedule anyway?" with "Yes, schedule anyway" and "No, go back to edit" buttons Click "Yes…" → Step 4; Click "No…" → restart Step 3

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
4b	Initiate+	Click “Save Draft Game to Post”	If 🔁Status = “Scheduled Draft” and “Modified by” = current User: Overwrite this record → Update Game record, “Last Modified” = Now, (keep current Version, Status, & “Modified by”)	Keep “Game” screen up
4c	Initiate+	Click “Save Draft Game to Post”	If 🔁Status = “Scheduled Draft” and “Modified by” <> current User: Create new record → Add Game record, “Last Modified” = Now, “Modified by” = User, Increment Version, (keep current Status and “Modified by”)	Keep “Game” screen up
4d	Initiate+	Click “Save Draft Game to Post”	If 🔁Status = “Scheduled”: Create new record → Add Game record, 🔁Status = “Scheduled Draft”, “Last Modified” = Now, “Modified by” = User (keep current Version)	Keep “Game” screen up
4e	Initiate+	Click “Save Game to Post” OR “Save Results Log”	If 🔁Status = “Scheduled” or “Logged” AND there are no changes: Do nothing	Keep “Game” screen up

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
4f	Initiate+	Click “Save Game to Post”	If 🔁Status = “Scheduled Draft” and “Modified by” = current User: Overwrite this record → Update Game record, 🔁Status = “Scheduled”, increment Version, “Last Modified” = Now, “Modified by” = User	Show “Game Management” screen (new Game is available)
4g	Initiate+	Click “Save Game to Post”	If 🔁Status = “Scheduled Draft” and “Modified by” <> current User: Create new record → Add Game record, 🔁Status = “Scheduled”, increment Version, “Last Modified” = Now, “Modified by” = User	Show “Game Management” screen (new Game is available)
4h	Initiate+	Click “Save Game to Post”	If 🔁Status = “Scheduled”: Create new record → Add Game record, 🔁Status = “Scheduled”, increment Version, “Last Modified” = Now, “Modified by” = User	Show “Game Management” screen (new Game is available)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
4g	Initiate+	Click “Save Draft Results Log”	If 🔁Status = “Scheduled” OR “Logged”: Create new record → Add Game record, 🔁Status = “Logged Draft”, “Last Modified” = Now, “Modified by” = User (keep current Version)	Keep “Game” screen up
4h	Initiate+	Click “Save Draft Results Log”	If 🔁Status = “Logged Draft” and “Modified by” = current User: Overwrite this record → Update Game record, “Last Modified” = Now, (keep current Version, Status, and “Modified by”)	Keep “Game” screen up
4i	Initiate+	Click “Save Draft Results Log”	If 🔁Status = “Logged Draft” and “Modified by” <> current User: Create new record → Add Game record, Increment Version, “Last Modified” = Now, “Modified by” = User (keep Status)	Keep “Game” screen up)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
4j	Initiate+	Click “Save Results Log”	If (🔁Status = “Scheduled” OR “Logged Draft”) AND “Modified by” == current User: Overwrite this record → Update Game record, 🔁Status = “Logged”, increment Version, “Last Modified” = Now, “Modified by” = User	Show “Game Management” screen (new Game is available)
4k	Initiate+	Click “Save Results Log”	If (🔁Status = “Scheduled” OR “Logged Draft”) AND “Modified by” <> current User: Create new record → Add Game record, 🔁Status = “Logged”, increment Version, “Last Modified” = Now, “Modified by” = User	Show “Game Management” screen (new Game is available)
4l	Initiate+	Click “Save Results Log”	If 🔁Status = “Logged” Create new record → Add Game record, 🔁Status = “Logged”, increment Version, “Last Modified” = Now, “Modified by” = User	Show “Game Management” screen (new Game is available)
5	System	Immediately after “Save Results Log” process	Process as applicable: Character’s Level Faction Level DM Variety Bonus	Send email to Character Owner with: Game Credit Rewards Character Milestone achieved (new Character Level, Faction Level, or DM Variety Bonus)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
6a	System	Nightly time-driven trigger	Scan for Games whose date was yesterday or earlier and 🔁Status = “Scheduled”	Email DM for Game: “Please log results”
6b	System	Nightly time-driven trigger	Scan for Games with: 🔁Status = “Scheduled Draft” OR 🔁Status = “Logged Draft”	Email DM for Game: “You have an unfinished draft {scheduled game}{game results log}.”

Character Management Workflow

Character attributes can be manually adjusted, but each “Save” or other trigger creates a new record, allowing for accountability when necessary.

Users can apply the DM Variety Bonus to any Character of their choosing, once available.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Player+	Menu: click “Character Management” under “Characters”	Read User profile	Show “Character Management” screen with Character list dropdown and “New Character” options
2	Player+	“Character Management”: select “New Character” or existing character from dropdowns. Filters available for DM+ who have access to other players’ characters	New Character: add Character record with “Draft” indication, “Version” set to 1; Existing Character selected: load Character data	Show “Character” screen with fields (editable)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
3	Player+	Fill in or edit fields, click “Save Draft” (optional)	If “Draft” record exists: update that record; If this updates an existing record: create new record, increment “Version” number, add “Draft” indication; If this creates new Character: create new record, set “Version” to 1, add “Draft” indication; In all cases: update “Last Modified” date and “Last Modified By” User.	Show “Character” screen with fields (editable) with “Draft saved” briefly displayed
4	Player+	Fill in or edit fields, click “Save”	If “Draft” record exists: update that record, remove “Draft” indication; If this updates an existing record: create new record, increment “Version” number; If this creates new Character: create new record, set “Version” to 1; In all cases: update “Last Modified” date and “Last Modified By” User.	Send email notification to User and (if different) User who owns edited/added Character

Character State Machine

…

Join Game Workflow

Players will join Games with chosen Characters from their lists. Doing this logs them as intending to attend a game, and makes their Player and Character information available for the Initiate+ to log the results.

There is no system check for Players signing up for multiple games at the same time. This will be managed in person.

Final attendance records are confirmed by the Initiate+ who runs the Game. It is the responsibility of the DM for the Game to ensure all Players submit their Characters in the system and to log the results. The result log confirms Player/Character attendance, game credit, faction credit, and rewards.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Player+	Menu: click “Upcoming Games” under “Games”	Read User profile, load Games, load Game Attendance	Show “Player Game Hub” screen with: If joined any pending Games: Show list of Games joined with “Leave Game” button next to each; Games list dropdown
2	Player+	“Player Game Hub”: select Game from dropdown	Load Game Report information	Display Game Report with “Join Game” button
3	Player+	“Game Report”: click “Join Game”	Load User’s Characters	Display Game Report (above) with the “Character Management” screen (embedded below): dropdown list of User’s Characters and a “New Character” button

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
4	Player+	Display Game Report (above) with Character Selection (below) (add a “Join with selected character” button)	See Character Management Workflow Steps 2-4 Upon Character selection OR completion of Character Management workflow and clicking “Join with selected character” → Add record to Game Attendance (GameAttendanceID, GameID, CharacterID, DropIn (default: “No”), GameCredit (blank), 🔁Status (“Joined”), Gold (blank), Items (blank), Notes (blank), DateCreated, CreatedBy (User))	Display Game Report (updated) with “Leave Game” button

Leave Game Workflow

If a User is not allowed to leave a particular Game, that Game will be filtered out and not displayed with the “Leave Game” option.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Player+	Menu: click “Upcoming Games” under “Games”	Read User profile, load Games, load Game Attendance; filter for available Games (this eliminates the possibility of leaving past or Logged games)	Show “Player Game Hub” screen with: If joined any pending Games: Show list of Games joined with “Leave Game” button next to each; Games list dropdown

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
2	Player+	“Player Game Hub”: click “Leave Game” → popup “Are you sure you want to withdraw from this game?” with “Yes” and “No” options → “No” returns to Step 1, “Yes” →	Update Game Attendance record 🔁Status = “Withdrew” & StatusChangeDate = Now	Show “Player Game Hub” screen with Games list dropdown

Create/Edit Faction Workflow

Faction Owners are DM+.

Faction Owners can start the creation process and save it as a draft if necessary. While in Draft status, it is not publicly viewable nor joinable.

Faction Owners and Mod+ can Archive a Faction. Archived Factions are publicly viewable but not joinable.

Faction 🔁Statuses: Draft, Active, Archive

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	DM+	Menu: click “Faction Management” under “Factions”	Read User profile, pull Factions information User has access to (DM+ can edit their own Factions; Mod+ can edit any Faction)	Show “Faction Management” screen with Factions list dropdown and “Create Faction” button
2	DM+	“Faction Management”: select from list of factions from dropdown or click “New Faction”	Load faction information or (if new) blank faction template	Show “Faction” screen with fields (editable)

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
3	DM+	Fill in or edit fields, click “Save Draft” (optional)	If draft exists: update that record; If this updates an existing Faction: add new Faction record, increment the version number, add “Draft” indication; If this is a new Faction: create a new record, set Version to 1, add “Draft” indication. In all cases: update “Last Modified” date and “Last Modified By” User.	Show “Faction” screen with fields (editable) with “Draft saved” displayed briefly
4	DM+	Fill in or edit fields, click “Save”	If draft exists: update that record, remove “Draft” indication; If this updates an existing Faction: add new Faction record, increment the version number; If this is a new Faction: create a new record, set Version to 1. In all cases: update “Last Modified” date and “Last Modified By” User.	Send email notification of Faction edit or add.
5	Faction Owner or Mod+	Click “Archive Faction” checkbox. A pop-up requests confirmation of this action before proceeding	Create new copy record; set 🔁Status = Archive	Show Faction screen in Archive status (still editable so that it can be un-Archived)

Join Faction Workflow

Leaving a Faction would be handled by the Faction Owner or Mod+ in Faction Management.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Player+	Menu: click “Join a Faction” under “Factions”	Read User profile, pull Characters list with associated Game information	Show “Join a Faction” screen with Factions list dropdown (filtered by possible options); If membership pending in any Factions, list them with date of request & Faction Owner’s name
2	Player+	“Join a Faction”: select Faction	Load Faction information	Display faction information; Display “Request to Join” button
3	Player+	“Join a Faction”: click “Request to Join”	Add record to “Faction Membership” (🔁Status = Requested; Date Requested)	Display Faction information with “Request Sent! You will be notified of the Faction Owner’s decision when it's made.” displayed. Email notification of request sent to Faction Owner
4	Faction Owner / Mod+	Faction Owner: Click “Manage Faction Membership” in email, OR Mod+: see Manage Faction Membership Workflow	Load Faction information	See Manage Faction Membership Workflow step 3

Manage Faction Membership Workflow

Faction Owners have a lot of power over the Faction membership. This workflow addresses any adjustments the Faction Owner (or Mod+) might make, including accepting or rejecting requests to join, confirming the Faction Level, and overriding their own rules for progression through the faction.

Faction Membership 🔁Statuses: Requested, Approved, Denied (initial status set by system when a Player requests a Faction; changes are managed by the Faction Owner or Mod+)

Faction Level: Managed by the system per rules established within the Faction table

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	Faction Owner	Menu: click “Faction Management” under “Factions”	Read User profile, pull Factions information User has access to	Show “Faction Management” screen with Factions list dropdown and “Create Faction” button
2	Faction Owner	“Faction Management”: select from list of factions from dropdown (filters available for “Owned” Factions)	Load Faction information	Display “Faction Membership” screen with “Faction” screen embedded above
3	Faction Owner	“Faction Management”: edit fields, select “Save” (Highlight “Status” fields = “Request Pending”) (Can override Member Level #)	Update Faction Membership information, including updating log for each Member, Date Joined (if applicable), and Last Modified Date	Display Faction information with Faction Membership information below (updated); send email notifications of changes to Members (only for accept/reject requests for membership and Level #)

Faction State Machine

Faction states are based on a set of rules established by the Faction Owner but within defined parameters.

- Faction Levels: minimum 3, maximum 5
- Promotion: require Faction Owner Approval (yes/no), minimum # Games in Faction (optional), minimum # Games in any Faction

Membership States: Requested → Approved

Level States: 1 → 2 → 3 [optionally: → 4 → 5]

[Requested] ——(Faction Owner/Mod+ approved)——→ [Approved]

[Level 1-4] ──(Faction credit received [guard:

Magic Item Workflow (TACBrew)

Standard Magic items will likely be managed directly in the database records as these won't change often and will require Mod+ approval.
TACBrew magic items can be submitted by any DM+ and currently have no approval required; however, we'll build an approval process for future use.

Step	Actor	UI Screen & Action	Data Read/Write	Next / Triggers
1	DM+			
2				
3				

TACBrew Magic Item State Machine

States: Draft, Submitted, Approved, Deprecated
DM+ can create a TACBrew magic item. It is in Draft state until submitted. Once reviewed, it is either Approved and available or Rejected and moved back to a Draft state. Users can delete their own TACBrew magic items that are in a Draft state, but only Mod+ can delete one in any other state.

Transitions:

- `User completes all required fields`: Draft state & system will allow the User to Submit the item
- `User submits item`: Draft → Submitted
- `Mod+ accepts item`: Submitted → Approved
- `Mod+ rejects item`: Submitted → Draft
- `Mod+ deprecates item`: Any State → Deprecated

```
[Draft]
  └──(User submits)──▸ [Submitted]
  ▲                          └──(Mod+ approves)──▸ [Approved]
  └──(Mod+ rejects)─────────┘
[any state] ──(Mod+ deprecates)──▸ [Deprecated]
```

Reports/Dashboards Workflow

Game Report

- Actor: Any User
- Menu: “Game Report” under “Games”
- Step 1: A Dropdown with list of games in descending order
- Step 2: Display Game Report for selected game with Game Name, Characters with their stats, and Party Composition information (Player Name only available to Player+)
- Party Composition Information: average character level, character level range, and experience points for various levels of play

Character Progress Report

- Actor: Player+
- Menu: “Character Progress Report” under “Characters”
- Step 1: Dropdown with a list of Characters filtered for the User (DM+ has access to all Characters, so add a “Player” filter (dropdown) for them so they can filter the Character list dropdown for just a specified User)
- Step 2: Character progress report for the selected Character

DM Games Dashboard

- Actor: DM+
- Menu: “DM Games Dashboard” under “Administration”
- One step: show dashboard (shows all DMs' status information, available to all DMs)

Pending Requests Report

This report is the only means of managing “stale” actions (besides daily email reminders for some pending actions). This report will be used to identify actions required by others and actions that need to be purged. Purging will be done manually, one item at a time, but we can develop a user interface that makes this relatively simple (a list with a button to “purge” or “delete” with a pop-up confirmation).

- Actor: DM+
- Menu: “Pending Requests Report” under “Administration”
- Step 1: Show the “Pending Requests Report” screen with links to each action (Initiate & DM only have access to their pending requests, Mod+ have access to all pending requests)
- Step 2: Clicking any action link opens up the screen where action is needed (if the User has access, else don’t display the link)

Administrative Meeting Workflow

This will be the same as the “Game Creation” and “Game Editing…” workflows, except it will display meeting information. Have a GameType field for “Game”, “Meeting”, and “Orientation”.

This can be used for Monthly DM Council Meetings, DM Orientation meetings, Lore Meetings, and so on.

There won’t be a “Join a Meeting” workflow, but we may add attendance records to be used by Mod+ if they request it later (this would require a new table, MeetingAttendance, with fields such as MeetingID, UserID, and so on; and new screens for Mods+ to add Users to meeting attendees list).

Password Reset Workflow

Let’s stick with just Google logins for now.

Archive Function (future add)

System action on a nightly time-driven trigger.

- Copy QuestBoard source Sheet for Full Archive.
- Keep the Full Archive copies from the last 7 days, and for the first of every month, purge the rest.
- Purge old records in the QuestBoard source Sheet:
    - Develop rules for each data table

Error Handling & Retries

Checks for failed emails or webhooks. TBD.

Step 2: Develop the Database Structure (Google Sheets)

Based on your workflows, you can now define the "tables" (which will be individual Google Sheets) and the "columns" (which will be the headers in those sheets). Think about the relationships between these sheets.

Once this table structure is developed, assess the scale at which this application is expected to grow within 6 months to determine whether Google Sheets remains a viable tool to use.

[Tables/list for each Table with field names, constraints (foreign keys, unique IDs), data types, validation rules…]

Step 3: Create Sheets, Populate with Test Data

Once you have these sheets set up in Google Sheets with appropriate headers and you've thought through the workflows, populate your sheets with some sample data. This will be invaluable for testing your Apps Script later.

Step 4: Create Web App Structure

Move to planning the web app's structure, which will involve:

1. **Apps Script Code.gs:** Your main server-side logic.
2. **HTML/CSS/JavaScript Files:** For your user interface. (PureCSS+)
3. **User Interface (UI) Flows:** What pages or views will each user role see? (e.g., DM Game Log Form, Player Character Sheet View, Admin Dashboard).
4. **Security Logic:** How will you enforce the login and role-based access using Apps Script?

Mobile-First Consideration: some actions (like “Log Results”) happen mid‑table; consider an “offline draft” mode or fast‑loading mobile view