

Feature Request System Specification

Overview

The Financial Playbook application includes a built-in **Feature Request System** (also known as a Feedback Portal or Idea Board) that allows users to submit ideas for new features, improvements, and bug reports. This system enables collaborative product development and ensures that the most valuable features are prioritized.

This is particularly valuable as the application expands from single-user to multi-user, allowing friends, family, and eventually paying customers to contribute to the product roadmap.

Core Features

1. Feature Request Submission

- Users can submit new feature ideas with detailed descriptions
- Categorization system (e.g., “Dashboard” , “QuickBooks Integration” , “Reports” , “UI/UX”)
- Priority indication (Nice to Have, Important, Critical)
- Attachment support (screenshots, mockups, documents)

2. Voting System

- Upvote/downvote functionality
- Vote count displayed prominently
- Users can see what they’ ve voted on
- Sorting by most popular requests

3. Status Tracking

- **Submitted** - New request awaiting review
- **Under Review** - Being evaluated by admin
- **Planned** - Approved and scheduled for development
- **In Progress** - Currently being worked on
- **Completed** - Feature has been implemented
- **Declined** - Request will not be implemented (with explanation)

4. Commenting & Discussion

- Users can comment on feature requests
- Admin can provide updates and ask clarifying questions
- Threaded discussions for better organization
- Email notifications for new comments

5. Public Roadmap

- Visual timeline showing planned features
- Categorized by release version or quarter
- Transparency into what's coming next

6. Admin Management

- Admin dashboard for reviewing and managing requests
 - Ability to change status and add admin notes
 - Merge duplicate requests
 - Analytics on request trends and popular categories
-

User Interface

Feature Request Board Page [QC: 900-950]

[201] [LOGO]

[202] Financial Playbook

[203] [Dashboard]

[204] [Accounts]

[205] [Scenarios]

[206] [Reports]

[222] [Feature Requests]

[207] [User: Dwain]

[208] [Logout]

[900] FEATURE REQUESTS

[901] [+ Submit New Request]

[902] FILTERS

[903] Status: [All ▼]

[904] Category: [All ▼]

[905] Sort: [Most Votes ▼]

[906] [Search requests...]

[907] POPULAR REQUESTS

[908] 🔥 45 votes

[909] PLANNED

[910] Add Plaid integration for additional bank accounts

[911] Submitted by: John D. | Category: Integrations

[912] 12 comments | Last updated: 2 days ago

[913] [👍 Upvote]

[914] [💬 Comment]

[915] [View Details]

[916] 🔥 38 votes

[917] IN PROGRESS

[918] Dark mode support for entire application

[919] Submitted by: Sarah M. | Category: UI/UX

[920] 8 comments | Last updated: 1 hour ago

[921] [👍 Upvote]

[922] [💬 Comment]

[923] [View Details]

[924] 🔥 27 votes

[925] UNDER REVIEW

[926] Export reports to Excel format

[927] Submitted by: You | Category: Reports

[928] 5 comments | Last updated: 3 days ago

[929] [👍 Upvote]

[930] [💬 Comment]

[931] [View Details]

[932] [Load More Requests]

Submit New Request Modal [QC: 940-960]

[940] Submit Feature Request [×]

[941] Title *

[942] [Enter a brief, descriptive title...]

[943] Category *

[944] [Select category ▼]

- Dashboard
- QuickBooks Integration
- Scenario Planning
- Reports
- Calendar
- UI/UX
- Security
- Other

[945] Priority

[946] ☐ Nice to Have ☐ Important ☐ Critical

[947] Description *

[948] [Describe the feature in detail...]
[Support for markdown formatting]

[949] Use Case (Optional)

[950] [Explain how this feature would help you...]

[951] Attachments (Optional)

[952] [Drag files here or click to upload]
Supported: PNG, JPG, PDF, up to 5MB

[953] [Submit Request] [954] [Cancel]

Feature Request Detail Page [QC: 960-990]

[201] [LOGO] [202] Financial Playbook [203] [Dashboard] [204]
[Accounts] [205] [Scenarios] [206] [Reports] |
[222] [Feature Requests] [207] [User: Dwain] [208]
[Logout]

[960] ← Back to Requests

[961] 🔥 45 votes [962] PLANNED
[963] Add Plaid integration for additional bank accounts

[964] Submitted by: John D. | Category: Integrations | Priority:
Important

[965] Created: Nov 15, 2025 | Last updated: Nov 21, 2025

[966] [👍 Upvote (45)] [967] [Share]

[968] DESCRIPTION
[969] It would be great to connect additional bank accounts beyond
QuickBooks using Plaid API. This would allow users to see
personal accounts alongside business accounts in one dashboard.

[970] USE CASE
[971] I have personal savings and investment accounts that impact my
overall financial picture but aren't in QuickBooks.

[972] ADMIN RESPONSE
[973] Great suggestion! We're evaluating Plaid integration for Q1 2026.
This would complement QuickBooks nicely. - Dwain (Nov 21, 2025)

[974] COMMENTS (12)

[975] Sarah M. - Nov 16, 2025
[976] +1 for this! I'd love to see my investment accounts too.
[977] [Reply]

[978] Mike T. - Nov 17, 2025
[979] Would this support credit unions? My primary account is with a

| local credit union.
| [980] [Reply]
|
| [981] [Load More Comments]
|
| [982] ADD COMMENT
| [983] [Write a comment...]
| [984] [Post Comment]
|

Public Roadmap Page [QC: 990-1020]

[201] [LOGO] [202] Financial Playbook [203] [Dashboard] [204]
[Accounts] [205] [Scenarios] [206] [Reports] |
[223] [Roadmap] [207] [User: Dwain] [208]
[Logout]

[990] PRODUCT ROADMAP

[991] Q1 2026 (January - March)

[992] ✓ COMPLETED

- [993] • Google Calendar Integration
- [994] • Visual Status System (Traffic Lights)

[995] 🛠 IN PROGRESS

- [996] • Dark Mode Support
- [997] • Customizable Dashboard Layouts

[998] 📅 PLANNED

- [999] • Plaid Integration for Additional Banks
- [1000] • Mobile App (iOS/Android)

[1001] Q2 2026 (April - June)

[1002] 📅 PLANNED

- [1003] • Multi-User Support (Team Collaboration)
- [1004] • Advanced Reporting with Custom Templates
- [1005] • Stripe Integration for Payment Processing

[1006] FUTURE (Under Consideration)

- [1007] • AI-Powered Financial Advisor Chatbot
- [1008] • Integration with Tax Software (TurboTax, etc.)
- [1009] • Cryptocurrency Portfolio Tracking

Admin Dashboard [QC: 1020-1050]

[\[201\]](#) [\[LOGO\]](#) [\[202\]](#) [Financial Playbook](#) [\[203\]](#) [\[Dashboard\]](#) [\[204\]](#)

[\[Accounts\]](#) [\[205\]](#) [\[Scenarios\]](#) [\[206\]](#) [\[Reports\]](#) |

[\[224\]](#) [\[Admin: Feature Requests\]](#) [\[207\]](#) [\[User: Dwain\]](#) [\[208\]](#)

[\[Logout\]](#)


[\[1020\]](#) [FEATURE REQUEST ADMIN](#)

[\[1021\]](#) [OVERVIEW](#)

[1022] Submitted 24	[1023] Under Review : 8	[1024] Planned 12
--	---	--

[1025] In Progress : 5	[1026] Completed : 47	[1027] Declined 15
--	---	---

[\[1028\]](#) [PENDING REVIEW \(8\)](#)

[1029]  15 votes Submitted: Nov 22, 2025
[1030] Export reports to Excel format
[1031] Category: Reports Priority: Important
[1032] [View Details] [1033] [Approve] [1034] [Decline]

[\[1035\]](#) [ANALYTICS](#)

[\[1036\]](#) [Most Requested Categories:](#)

- [Integrations](#) (32%)
- [Reports](#) (24%)
- [UI/UX](#) (18%)
- [Dashboard](#) (14%)
- [Other](#) (12%)

Database Schema

```
// Feature requests table
export const featureRequests = mysqlTable("feature_requests", {
  id: int("id").autoincrement().primaryKey(),
  userId: int("user_id").notNull(), // Submitter
  title: varchar("title", { length: 255 }).notNull(),
  description: text("description").notNull(),
  useCase: text("use_case"), // Optional
  category: mysqlEnum("category", [
    "dashboard",
    "quickbooks",
    "scenarios",
    "reports",
    "calendar",
    "ui_ux",
    "security",
    "other"
  ]).notNull(),
  priority: mysqlEnum("priority", ["nice_to_have", "important",
"critical"]).default("nice_to_have"),
  status: mysqlEnum("status", [
    "submitted",
    "under_review",
    "planned",
    "in_progress",
    "completed",
    "declined"
  ]).default("submitted"),
  voteCount: int("vote_count").default(0),
  commentCount: int("comment_count").default(0),
  adminNotes: text("admin_notes"), // Internal notes for admin
  adminResponse: text("admin_response"), // Public response to users
  completedAt: timestamp("completed_at"),
  declinedReason: text("declined_reason"),
  createdAt: timestamp("created_at").defaultNow().notNull(),
  updatedAt: timestamp("updated_at").defaultNow().onUpdateNow().notNull(),
});

// Votes table
export const featureRequestVotes = mysqlTable("feature_request_votes", {
  id: int("id").autoincrement().primaryKey(),
  requestId: int("request_id").notNull(), // Foreign key to feature_requests
  userId: int("user_id").notNull(), // Voter
```

```

    voteType: mysqlEnum("vote_type", ["upvote", "downvote"]).notNull(),
    createdAt: timestamp("created_at").defaultNow().notNull(),
  });

// Comments table
export const featureRequestComments = mysqlTable("feature_request_comments", {
  id: int("id").autoincrement().primaryKey(),
  requestId: int("request_id").notNull(), // Foreign key to feature_requests
  userId: int("user_id").notNull(), // Commenter
  parentCommentId: int("parent_comment_id"), // For threaded replies
  content: text("content").notNull(),
  createdAt: timestamp("created_at").defaultNow().notNull(),
  updatedAt: timestamp("updated_at").defaultNow().onUpdateNow().notNull(),
});

// Attachments table
export const featureRequestAttachments =
mysqlTable("feature_request_attachments", {
  id: int("id").autoincrement().primaryKey(),
  requestId: int("request_id").notNull(), // Foreign key to feature_requests
  fileName: varchar("file_name", { length: 255 }).notNull(),
  fileUrl: varchar("file_url", { length: 500 }).notNull(),
  fileSize: int("file_size").notNull(), // In bytes
  mimeType: varchar("mime_type", { length: 100 }).notNull(),
  uploadedAt: timestamp("uploaded_at").defaultNow().notNull(),
});

```


Key Workflows

Submit New Feature Request

1. User clicks [+ Submit New Request] button
2. Modal opens with submission form
3. User fills in:
 - Title (required)
 - Category (required)
 - Priority (optional, defaults to “Nice to Have”)

- Description (required, markdown supported)
 - Use Case (optional)
 - Attachments (optional, up to 5MB)
4. User clicks [Submit Request]
 5. System validates input
 6. Request is created with status “Submitted”
 7. Email notification sent to admin
 8. User is redirected to request detail page
 9. Success toast: “Your feature request has been submitted!”

Vote on Feature Request

1. User browses feature request board
2. User clicks [ Upvote] on a request
3. Vote count increments immediately (optimistic update)
4. Vote is saved to database
5. User can click again to remove vote
6. User can only vote once per request

Admin Reviews Request

1. Admin logs in and navigates to Admin Dashboard
2. Admin sees list of requests “Under Review”
3. Admin clicks [View Details] on a request
4. Admin reads description, use case, and comments
5. Admin decides to approve or decline
6. If **Approve**:
 - Admin changes status to “Planned”
 - Admin adds admin response explaining timeline
 - Admin adds request to roadmap
7. If **Decline**:

- Admin changes status to “Declined”
 - Admin provides decline reason
8. Email notification sent to submitter
 9. Request appears in appropriate section of board

Feature Completion

1. Admin marks feature as “In Progress” when development starts
 2. Admin marks feature as “Completed” when deployed
 3. Email notification sent to all users who voted or commented
 4. Request moves to “Completed” section
 5. Request appears on roadmap with ✓ checkmark
-

Email Notifications

New Feature Request Submitted (to Admin)

Subject: New Feature Request: [Title]

A new feature request has been submitted:

Title: Add Plaid integration for additional bank accounts

Category: Integrations

Priority: Important

Submitted by: John D.

[\[View Request\]](#) [\[Approve\]](#) [\[Decline\]](#)

Feature Status Changed (to Submitter & Voters)

Subject: Feature Request Update: [Title]

The feature request you submitted/voted **on** has been updated:

Title: Add Plaid integration **for** additional bank accounts

Status: Submitted → Planned

Admin Response:

Great suggestion! We're evaluating Plaid integration for Q1 2026.
This would complement QuickBooks nicely.

[\[View Request\]](#)

Feature Completed (to Submitter & Voters)

Subject: Feature Request Completed: [Title]

Good news! The feature you requested has been completed:

Title: Dark Mode Support

Status: Completed

Your idea **is** now live **in** the application. Thank you **for** helping
improve Financial Playbook!

[\[Try It Now\]](#)

Analytics & Insights

Admin Dashboard Metrics

- Total requests by status
- Most popular categories
- Average time from submission to completion

- User engagement (votes, comments)
- Top contributors (most requests submitted)

Trending Requests

- Requests with most votes in last 7 days
 - Requests with most comments in last 7 days
 - Fastest-growing requests (vote velocity)
-

Implementation Priority

Phase 1 (MVP)

- ☐ Basic feature request submission
- ☐ Feature request list view
- ☐ Voting system (upvote only)
- ☐ Status tracking
- ☐ Admin approval/decline

Phase 2

- ☐ Commenting system
- ☐ Email notifications
- ☐ Public roadmap view
- ☐ Filtering and search

Phase 3

- ☐ Attachment support
- ☐ Threaded comments
- ☐ Analytics dashboard
- ☐ Request merging

Benefits

For Users

- Direct channel to influence product direction
- Transparency into what's being worked on
- Community engagement and validation of ideas
- Sense of ownership in the product

For Developer (You)

- Prioritize features based on real user demand
- Reduce support burden (users can see if feature is planned)
- Build community and user loyalty
- Gather detailed requirements from users

For Future Business

- Validate market demand before building
 - Marketing content (show active development)
 - Competitive advantage (responsive to users)
 - Potential for premium features (most-voted paid features)
-

Accessibility & Best Practices

- Keyboard navigation for voting and commenting
- Screen reader support for status changes
- Clear visual indicators for voted requests
- Mobile-responsive design
- Rate limiting to prevent spam

- Moderation tools for inappropriate content
-

Testing Checklist

- ☐ Users can submit feature requests
 - ☐ Voting system works correctly
 - ☐ Vote counts update in real-time
 - ☐ Users can only vote once per request
 - ☐ Comments can be added and displayed
 - ☐ Admin can change request status
 - ☐ Email notifications are sent correctly
 - ☐ Filtering and search work as expected
 - ☐ Roadmap displays planned features
 - ☐ Analytics dashboard shows accurate data
 - ☐ Mobile responsiveness works
 - ☐ Accessibility features work with screen readers
-

Document Version: 1.0

Last Updated: November 23, 2025

Related: Financial Playbook Complete Documentation v8.0