# HOA Management System Integration Test Plan

HOA\_MANAGEMENT\_INT\_TP\_01

Version: 1.0 Last Updated: November 28, 2024

## Part 1: Authentication Flow Tests

### HOA-INT-IT-AUTH-001: Complete Login Process

Priority: High Components: Login.js, AuthContext.js, apiService.js

**Test Steps:** 1. Navigate to login page 2. Enter test credentials: - Email: Ada.Nolcrest@coldmail.com - Password: Pas$w0rd1! 3. Click login button 4. Monitor network request in DevTools 5. Check localStorage for token 6. Verify dashboard redirect 7. Check navigation menu updates 8. Verify user info display

**Expected Results:** - Login form displays correctly - Credentials accepted - Network request shows 200 status - Token stored in localStorage - Successful redirect to dashboard - User menu shows logged-in state - Protected routes accessible

**Visual Verification Points:** - Login form field validation - Loading state during submission - Smooth transition to dashboard - Updated navigation items - User profile information display

### HOA-INT-IT-AUTH-002: Board Login Process

Priority: High Components: Login.js, AuthContext.js, apiService.js

**Test Steps:** 1. Log in with board member credentials 2. Monitor permission loading 3. Verify board navigation items appear 4. Check board dashboard access 5. Test board-specific actions 6. Verify permission-based UI elements 7. Check role-based content display

**Expected Results:** - Login successful with board credentials - Board member permissions loaded - Additional navigation items visible - Board dashboard accessible - Board actions enabled - Role-specific UI elements shown - Board content properly displayed

**Visual Verification Points:** - Board member UI differences - Permission-based button states - Board action menu items - Role indicator in profile

### HOA-INT-IT-AUTH-003: First Time Login Process

Priority: High Components: Login.js, ChangePasswordModal.js, apiService.js

**Test Steps:** 1. Log in with temporary credentials 2. Verify password change prompt 3. Test password requirements: - Minimum length - Special characters - Numbers - Case mixing 4. Enter new password 5. Confirm new password 6. Submit password change 7. Verify re-authentication 8. Check dashboard access

**Expected Results:** - Temporary password accepted - Password change modal appears - Password requirements displayed - Validation feedback shown - Change successful - Auto re-authentication - Dashboard access granted

**Visual Verification Points:** - Password modal display - Requirement indicators - Validation messages - Success confirmation - Session maintenance

### HOA-INT-IT-AUTH-004: New Registration Process

Priority: High Components: RegistrationModal.js, apiService.js

**Test Steps:** 1. Enter registration details: - Account ID - Owner ID - Temporary code 2. Submit initial verification 3. Complete personal info: - First name - Last name 4. Enter contact details: - Email - Phone 5. Set password: - New password - Confirmation 6. Verify email notification 7. Test immediate login

**Expected Results:** - Registration form loads - Code verification works - Personal info saved - Contact details validated - Password requirements met - Email notification sent - Login enabled after completion

**Visual Verification Points:** - Multi-step form progress - Field validation states - Error messages - Success indicators - Email preview

### HOA-INT-IT-AUTH-005: Registration Code Verification

Priority: High Components: RegistrationModal.js, apiService.js

**Test Steps:** 1. Enter registration code 2. Test invalid codes: - Wrong format - Expired code - Used code 3. Enter valid code 4. Check verification status 5. Monitor API response 6. Verify next step access 7. Test code reuse prevention

**Expected Results:** - Code input works - Invalid codes rejected - Valid code accepted - Verification confirmed - Next step enabled - Used code blocked - Error messages shown

**Visual Verification Points:** - Code input format - Validation feedback - Error states - Success indication - Progress update

### HOA-INT-IT-AUTH-006: Password Reset Request Flow

Priority: High Components: ForgotPasswordModal.js, apiService.js

**Test Steps:** 1. Access forgot password 2. Enter email address 3. Submit reset request 4. Check email delivery 5. Verify reset token 6. Test invalid emails: - Non-existent - Invalid format - Inactive accounts 7. Monitor rate limiting

**Expected Results:** - Reset form displays - Email validation works - Request processed - Email sent - Token generated - Invalid emails handled - Rate limits enforced

**Visual Verification Points:** - Form display - Validation messages - Success confirmation - Email template - Error states

### HOA-INT-IT-AUTH-007: Password Reset Completion

Priority: High Components: ForgotPasswordModal.js, apiService.js

**Test Steps:** 1. Access reset link 2. Verify token validity 3. Enter new password 4. Confirm new password 5. Submit password change 6. Test invalid scenarios: - Expired token - Used token - Password mismatch 7. Verify login access

**Expected Results:** - Reset form loads - Token verified - Password updated - Invalid cases handled - Login enabled - Old password invalid - Session updated

**Visual Verification Points:** - Reset form display - Validation feedback - Success message - Login redirect - Error handling

### HOA-INT-IT-AUTH-008: Token Refresh Process

Priority: High Components: AuthContext.js, apiService.js

**Test Steps:** 1. Monitor token expiration 2. Check refresh trigger 3. Verify new token 4. Test session continuity 5. Check request headers 6. Verify storage update 7. Monitor refresh timing

**Expected Results:** - Refresh triggers properly - New token generated - Session maintained - Headers updated - Storage updated - Timing correct - No session interruption

**Visual Verification Points:** - No UI interruption - Session maintenance - Storage updates - Network requests

### HOA-INT-IT-AUTH-009: Session Expiration Handling

Priority: High Components: AuthContext.js, apiService.js

**Test Steps:** 1. Let session expire 2. Monitor expiration event 3. Check cleanup process 4. Verify redirect 5. Test API responses 6. Check storage cleanup 7. Verify re-login requirement

**Expected Results:** - Expiration detected - Session cleaned - Storage cleared - User redirected - Re-login required - Data protected - State reset

**Visual Verification Points:** - Expiration notice - Cleanup feedback - Redirect handling - Login requirement

### HOA-INT-IT-AUTH-010: Complete Logout Process

Priority: High Components: AuthContext.js, apiService.js

**Test Steps:** 1. Initiate logout 2. Check token removal 3. Verify storage cleanup 4. Monitor state reset 5. Test route protection 6. Check API access 7. Verify redirect

**Expected Results:** - Logout processes - Token removed - Storage cleared - State reset - Routes protected - API access blocked - Redirect to login

**Visual Verification Points:** - Logout confirmation - Cleanup feedback - Redirect handling - Protected route blocks

## 

## Part 2: Payment Flow Tests

### HOA-INT-IT-PAY-001: New Payment Process

Priority: High Components: PaymentModal.js, apiService.js

**Test Steps:** 1. Navigate to Payments section 2. Click “Make Payment” button 3. Enter payment details: - Amount validation - Minimum payment check - Maximum payment check - Decimal handling 4. Select existing payment method 5. Review payment summary 6. Submit payment 7. Monitor processing state 8. Verify payment confirmation 9. Check payment history update 10. Verify receipt generation

**Expected Results:** - Payment modal opens correctly - Amount validation works - Payment method selection functions - Processing indicator shows - Confirmation displays - Receipt generates - History updates immediately - Balance updates correctly

**Visual Verification Points:** - Modal rendering - Form validation states - Processing animation - Success confirmation - Receipt format - Updated balance display

### HOA-INT-IT-PAY-002: Payment with New Card

Priority: High Components: PaymentModal.js, CardManagementModal.js

**Test Steps:** 1. Start payment process 2. Select “Add New Card” 3. Enter card details: - Card number - Expiration date - CVV - Billing name 4. Test card validation 5. Save card for future use option 6. Complete full payment process 7. Verify card saved 8. Check payment success 9. Verify card appears in saved cards

**Expected Results:** - Card form displays properly - Validation works for all fields - Card saves if selected - Payment processes - New card appears in list - Receipt shows new card - Payment history updated

**Visual Verification Points:** - Card form display - Validation messages - Save card checkbox - Processing state - Success confirmation - Card list update

### HOA-INT-IT-PAY-003: Failed Payment Handling

Priority: High Components: PaymentModal.js, apiService.js

**Test Steps:** 1. Attempt payment with: - Invalid card - System error simulation 2. Monitor error handling 3. Verify error messages 4. Check retry options 5. Test alternative payment 6. Verify no double charges 7. Check failed payment logging

**Expected Results:** - Error caught properly - Clear error message shown - Retry option available - No duplicate charges - Session maintained - Balance unchanged - Error logged correctly

**Visual Verification Points:** - Error message display - Retry button state - Form reset state - Balance display - History update

### HOA-INT-IT-PAY-004: Card Management - Add Card

Priority: High Components: CardManagementModal.js, apiService.js

**Test Steps:** 1. Open card management 2. Click “Add Card” 3. Enter test card details: - Multiple card types - Various number formats - Different expiration dates 4. Test validation rules 5. Save new card 6. Verify card listing 7. Test default card setting

**Expected Results:** - Add card form opens - Validation works - Card saves successfully - List updates - Default option works - Card usable for payments - Multiple cards supported

**Visual Verification Points:** - Form rendering - Validation states - Success message - Card list update - Default card indicator

### HOA-INT-IT-PAY-005: Card Management - Remove Card

Priority: High Components: CardManagementModal.js, apiService.js

**Test Steps:** 1. Open card management 2. Select card to remove 3. Confirm deletion 4. Test deletion scenarios: - Default card - Recently used card - Single card removal 5. Verify card removal 6. Check payment options update 7. Test payment flow after removal

**Expected Results:** - Deletion confirmation shows - Card removes successfully - Default card handling works - Payment options update - No orphaned references - History maintained - System stability preserved

**Visual Verification Points:** - Confirmation modal - Success message - Updated card list - Payment form update - History display

### HOA-INT-IT-PAY-006: Default Card Management

Priority: High Components: CardManagementModal.js, apiService.js

**Test Steps:** 1. View saved cards 2. Set default card: - From card list - During new card add - After payment 3. Change default card 4. Verify payment defaults 5. Test removal impact 6. Check payment form behavior 7. Verify persistence

**Expected Results:** - Default setting works - Selection persists - Payment form updates - Removal handled - New default selection - User preferences saved - Consistent behavior

**Visual Verification Points:** - Default indicator - Selection UI - Payment form default - Update confirmation - Preference persistence

### HOA-INT-IT-PAY-007: Receipt Generation Process

Priority: High Components: PaymentHistoryModal.js, apiService.js

**Test Steps:** 1. Complete payment 2. Generate receipt: - Automatic generation - Manual request 3. Verify receipt content: - Payment details - Account info - Date/time - Transaction ID 4. Test download options 5. Check email delivery 6. Verify PDF format 7. Test print functionality

**Expected Results:** - Receipt generates automatically - Manual generation works - All details included - Download functions - Email sends - PDF formats correctly - Print layout proper

**Visual Verification Points:** - Receipt display - Download options - Email format - PDF layout - Print preview

### HOA-INT-IT-PAY-008: Receipt Download Functionality

Priority: High Components: PaymentHistoryModal.js, apiService.js

**Test Steps:** 1. Access payment history 2. Select receipt 3. Test download options: - PDF format - Print version - Email copy 4. Verify file generation 5. Check file format 6. Test multiple downloads 7. Verify file naming

**Expected Results:** - Download initiates - File generates correctly - Format options work - Files named properly - Email delivers - Print formats - Multiple downloads work

**Visual Verification Points:** - Download buttons - Progress indicator - Success message - File naming - Format options

### HOA-INT-IT-PAY-009: Payment History Real-time Updates

Priority: High Components: Payments.js, apiService.js

**Test Steps:** 1. Monitor payment list 2. Complete new payment 3. Verify instant update 4. Check sort order 5. Verify payment details 6. Test filter updates 7. Check balance reflection

**Expected Results:** - List updates instantly - Sort maintains - Details accurate - Filters work - Balance updates - History accurate - UI consistent

**Visual Verification Points:** - List animation - Update timing - Sort indication - Filter state - Balance display

### HOA-INT-IT-PAY-010: Payment History Filter/Search

Priority: High Components: Payments.js, apiService.js

**Test Steps:** 1. Access payment history 2. Test filters: - Date range - Amount range - Payment type - Status 3. Use search function 4. Apply multiple filters 5. Clear filters 6. Test sort with filters 7. Verify result accuracy

**Expected Results:** - Filters apply correctly - Search functions - Multiple filters work - Clear resets view - Sort maintains - Results accurate - UI responsive

**Visual Verification Points:** - Filter controls - Search input - Results display - Clear indication - Sort indicators

## 

## Part 3: Document Flow Tests

### HOA-INT-IT-DOC-001: Single Document Upload Process

Priority: High Components: UploadDocumentModal.js, apiService.js

**Test Steps:** 1. Navigate to Documents section 2. Click “Upload Document” button 3. Test file selection: - Select single document - Verify allowed file types - Check file size limits 4. Enter document metadata: - Title - Description - Category 5. Submit upload 6. Monitor upload progress 7. Verify success notification 8. Check document list update 9. Verify file accessibility

**Expected Results:** - Upload modal opens properly - File selection works - Metadata form validates - Upload progress shows - Success notification appears - Document list updates - File accessible after upload - Categories update correctly

**Visual Verification Points:** - Modal rendering - Upload progress bar - Success message - List update animation - Document card display

### HOA-INT-IT-DOC-002: Multiple Document Upload

Priority: High Components: UploadDocumentModal.js, apiService.js

**Test Steps:** 1. Access upload interface 2. Select multiple files: - Different file types - Various sizes - Multiple categories 3. Enter batch metadata 4. Monitor upload queue 5. Track progress indicators 6. Verify success rate 7. Check error handling 8. Verify document list

**Expected Results:** - Multiple selection works - Queue displays properly - Progress tracks accurately - Successful files upload - Failed files noted - List updates completely - Error handling works - Status updates show

**Visual Verification Points:** - Queue display - Progress indicators - Status updates - Error messages - List refresh

### HOA-INT-IT-DOC-003: Large File Upload Handling

Priority: High Components: UploadDocumentModal.js, apiService.js

**Test Steps:** 1. Select large file(s) 2. Check size validation 3. Monitor upload progress 4. Test connection interruption 5. Verify resume capability 6. Check timeout handling 7. Verify completion 8. Test failed upload recovery 9. Check system performance 10. Verify file integrity

**Expected Results:** - Size limits enforced - Progress shows accurately - Interruption handled - Resume works - Timeouts managed - Completion successful - Recovery functions - Performance maintained - Files verify correctly

**Visual Verification Points:** - Size warnings - Progress bar - Interruption notice - Resume button - Success/failure states

### HOA-INT-IT-DOC-004: Single Document Download Process

Priority: High Components: Documents.js, apiService.js

**Test Steps:** 1. Locate document in list 2. Click download button/icon 3. Monitor download initiation: - Browser download prompt - File name verification - Size indication 4. Track download progress 5. Verify file completion 6. Check downloaded file: - Name matches - Size correct - Can be opened 7. Test different file types 8. Verify download history

**Expected Results:** - Download initiates properly - Progress indicator shows - File saves correctly - Name preserved - File opens properly - All types download - History tracks downloads - System remains responsive

**Visual Verification Points:** - Download button state - Progress indicator - Success notification - File location prompt - History update

### HOA-INT-IT-DOC-005: Batch Document Download

Priority: High Components: Documents.js, apiService.js

**Test Steps:** 1. Select multiple documents 2. Initiate batch download 3. Verify compression process 4. Monitor download progress: - Overall progress - Individual file status 5. Check zip file creation 6. Verify file organization 7. Test extraction 8. Verify file integrity 9. Check large batch handling

**Expected Results:** - Multiple selection works - Batch process starts - Zip creates properly - Progress shows accurately - Files organize correctly - All files complete - Extraction works - Files verify intact - Large batches handle

**Visual Verification Points:** - Selection interface - Progress indication - Status updates - Completion notice - File structure

### HOA-INT-IT-DOC-006: Document Metadata Update

Priority: High Components: DocumentCard.js, apiService.js

**Test Steps:** 1. Select document to edit 2. Open edit interface 3. Modify metadata: - Title change - Description update - Category adjustment 4. Save changes 5. Verify updates 6. Check history 7. Test multiple updates 8. Verify search impact

**Expected Results:** - Edit mode opens - Changes save properly - Updates reflect immediately - History tracks changes - Search indexes update - Multiple edits work - System remains stable - UI updates correctly

**Visual Verification Points:** - Edit form display - Save button state - Update animation - Success message - List refresh

### HOA-INT-IT-DOC-007: Document File Replacement

Priority: High Components: DocumentCard.js, apiService.js

**Test Steps:** 1. Select document 2. Choose replace option 3. Select new file: - Same type - Different type - Various sizes 4. Verify file validation 5. Upload replacement 6. Check version handling 7. Verify metadata retention 8. Test access continuity

**Expected Results:** - Replace option works - Validation functions - Upload completes - Versions track properly - Metadata preserves - Access maintains - UI updates - Links update

**Visual Verification Points:** - Replace interface - Progress indicator - Version display - Success message - Access indicators

### HOA-INT-IT-DOC-008: Single Document Deletion

Priority: High Components: DocumentCard.js, apiService.js

**Test Steps:** 1. Select document to delete 2. Trigger deletion 3. Confirm deletion intent 4. Monitor deletion process 5. Verify removal: - List update - Storage cleanup - Link removal 6. Check related items 7. Test undo capability 8. Verify system impact

**Expected Results:** - Delete option shows - Confirmation displays - Process completes - List updates properly - Storage cleans - Links remove - Related items update - System stable

**Visual Verification Points:** - Delete button - Confirmation modal - Progress indicator - List update - Success message

### HOA-INT-IT-DOC-009: Batch Document Deletion

Priority: High Components: DocumentCard.js, apiService.js

**Test Steps:** 1. Select multiple documents 2. Initiate batch deletion 3. Review selection summary 4. Confirm deletion 5. Monitor batch process: - Progress tracking - Error handling 6. Verify complete removal 7. Check system resources 8. Test partial failures

**Expected Results:** - Multiple select works - Summary displays - Confirmation shows - Progress tracks - Errors handle properly - Removal completes - Resources free - Partial process works

**Visual Verification Points:** - Selection interface - Confirmation display - Progress tracking - Error messages - Success indication

### HOA-INT-IT-DOC-010: Document Search Functionality

Priority: High Components: Documents.js, apiService.js

**Test Steps:** 1. Access search interface 2. Test search criteria: - Title search - Content search - Category filter - Date range 3. Enter search terms 4. Monitor result loading 5. Test result filtering 6. Verify result accuracy 7. Check performance 8. Test edge cases

**Expected Results:** - Search interface works - Filters function - Results load properly - Accuracy maintains - Performance good - Edge cases handle - UI responsive - Results clear

**Visual Verification Points:** - Search interface - Filter controls - Results display - Loading indicator - No results state

## 

## Part 4: Message Flow Tests

### HOA-INT-IT-MSG-001: New Message Creation

Priority: High Components: NewMessageModal.js, apiService.js

**Test Steps:** 1. Click “New Message” button 2. Test recipient selection: - Search functionality - Multiple recipients - Invalid selections 3. Enter message content: - Text formatting - Length limits - Required fields 4. Test draft saving 5. Submit message 6. Verify sending status 7. Check sent folder 8. Monitor recipient notification

**Expected Results:** - Modal opens correctly - Recipient search works - Content saves properly - Draft saves function - Message sends successfully - Sent folder updates - Notifications trigger - UI remains responsive

**Visual Verification Points:** - Compose modal display - Recipient list format - Sending indicator - Success notification - Sent folder update

### HOA-INT-IT-MSG-002: Message with Attachment

Priority: High Components: NewMessageModal.js, apiService.js

**Test Steps:** 1. Create new message 2. Add attachments: - Single file - Multiple files - Size limits - File types 3. Test attachment preview 4. Monitor upload progress 5. Compose message text 6. Submit with attachments 7. Verify complete delivery 8. Check attachment access

**Expected Results:** - Attachment selection works - Size limits enforce - File types validate - Preview displays - Upload progress shows - Message sends complete - Attachments deliver - Access permissions set

**Visual Verification Points:** - Attachment interface - Upload progress - Preview display - Size warnings - Success states

### HOA-INT-IT-MSG-003: Direct Message Reply

Priority: High Components: ReplyMessageModal.js, apiService.js

**Test Steps:** 1. Open received message 2. Click reply button 3. Verify pre-filled fields: - Recipient - Subject - Quote original 4. Add reply content 5. Test formatting 6. Submit reply 7. Check threading 8. Verify notifications

**Expected Results:** - Reply modal opens - Fields pre-fill correctly - Formatting works - Reply sends properly - Thread updates - Notifications send - History maintains - UI updates properly

**Visual Verification Points:** - Reply form display - Quoted content format - Send button state - Thread view update - Success message

### HOA-INT-IT-MSG-004: Thread Reply Process

Priority: High Components: ReplyMessageModal.js, apiService.js

**Test Steps:** 1. Open message thread 2. Select reply point 3. Test reply options: - Reply to all - Single reply - Forward 4. Check recipient handling 5. Add reply content 6. Submit threaded reply 7. Verify thread update 8. Check notification flow

**Expected Results:** - Thread view displays - Reply options work - Recipients handle correctly - Content adds properly - Thread updates - Notifications send - History maintains - UI refreshes

**Visual Verification Points:** - Thread display - Reply options - Recipient list - Thread structure - Update animation

### HOA-INT-IT-MSG-005: Message List Loading

Priority: High Components: Messages.js, apiService.js

**Test Steps:** 1. Access message inbox 2. Monitor initial load 3. Test filtering: - By date - By status - By type 4. Check pagination 5. Test refresh function 6. Verify sort options 7. Check unread indicators 8. Test bulk actions

**Expected Results:** - List loads properly - Filters work correctly - Pagination functions - Refresh updates - Sort maintains - Indicators show - Bulk actions work - Performance maintains

**Visual Verification Points:** - Loading state - Filter controls - Page navigation - Sort indicators - Unread markers

### HOA-INT-IT-MSG-006: Thread View Display

Priority: High Components: MessageDetail.js, apiService.js

**Test Steps:** 1. Select conversation thread 2. Monitor thread loading 3. Test thread interactions: - Expand/collapse - Load more - Jump to message 4. Verify chronological order 5. Check participant list 6. Test inline actions 7. Verify media display 8. Check thread navigation

**Expected Results:** - Thread loads completely - Interactions work - Order maintains - Participants show - Actions function - Media displays - Navigation works - Performance good

**Visual Verification Points:** - Thread structure - Expansion controls - Participant display - Media previews - Navigation elements

### HOA-INT-IT-MSG-007: Message Recipient Search

Priority: High Components: NewMessageModal.js, apiService.js

**Test Steps:** 1. Open recipient search 2. Test search criteria: - Name search - Role filter - Property filter 3. Enter search terms 4. Check search results 5. Test selection process 6. Verify multiple select 7. Test invalid selections 8. Check performance

**Expected Results:** - Search opens properly - Filters apply - Results display - Selection works - Multiple select functions - Validation works - Performance maintains - UI responsive

**Visual Verification Points:** - Search interface - Results display - Selection indicators - Filter controls - Loading states

### HOA-INT-IT-MSG-008: Message Content Search

Priority: High Components: Messages.js, apiService.js

**Test Steps:** 1. Access search function 2. Test search parameters: - Content search - Date range - Message type - Attachment presence 3. Enter search terms 4. Monitor result loading 5. Check result accuracy 6. Test advanced filters 7. Verify performance 8. Test result actions

**Expected Results:** - Search interface works - Parameters apply - Results load properly - Accuracy maintains - Filters function - Actions available - Performance good - UI responsive

**Visual Verification Points:** - Search controls - Results format - Filter display - Loading indicator - Action buttons

### HOA-INT-IT-MSG-009: Message Deletion Process

Priority: High Components: MessageDetail.js, apiService.js

**Test Steps:** 1. Select message(s) 2. Initiate deletion 3. Confirm deletion intent 4. Monitor deletion process 5. Test bulk deletion: - Single message - Multiple messages - Entire thread 6. Check thread impact 7. Verify cleanup 8. Test recovery options

**Expected Results:** - Selection works - Confirmation shows - Process completes - Bulk delete functions - Threads update - Cleanup successful - Recovery available - UI updates properly

**Visual Verification Points:** - Delete controls - Confirmation modal - Progress indicator - Success message - Thread updates

### HOA-INT-IT-MSG-010: Thread Deletion Process

Priority: High Components: MessageDetail.js, apiService.js

**Test Steps:** 1. Select message thread 2. Initiate thread deletion 3. Review deletion scope: - All messages - Attachments - References 4. Confirm deletion 5. Monitor batch process 6. Verify complete removal 7. Check related items 8. Test partial deletion

**Expected Results:** - Thread selection works - Scope shows clearly - Confirmation displays - Process tracks - Removal complete - Relations update - Partial works - UI refreshes

**Visual Verification Points:** - Selection interface - Scope display - Progress tracking - Success message - List refresh

## 

## Part 5: Board Member Flow Tests

### HOA-INT-IT-BRD-001: Board Role Creation

Priority: High Components: BoardMemberSettingsModal.js

**Test Steps:** 1. Access board settings 2. Click “Add New Role” 3. Enter role details: - Role title - Description - Permission sets: - Assess fines - Change rates - Change members 4. Save new role 5. Verify role creation 6. Test role permissions 7. Check role listing 8. Test duplicate prevention

**Expected Results:** - Settings modal opens - Form validates input - Permissions save - Role creates successfully - Listing updates - Permissions work - Duplicates prevented - UI updates properly

**Visual Verification Points:** - Role form display - Permission toggles - Success message - Role list update - Error states

### HOA-INT-IT-BRD-002: Role Permission Update

Priority: High Components: BoardMemberSettingsModal.js

**Test Steps:** 1. Select existing role 2. Open edit interface 3. Modify permissions: - Toggle settings - Update access levels - Change descriptions 4. Save changes 5. Verify updates 6. Test affected users 7. Check audit trail 8. Verify system impact

**Expected Results:** - Edit mode opens - Changes save - Updates apply - Users update - Audit logs - System stable - UI refreshes - History maintains

**Visual Verification Points:** - Edit interface - Permission controls - Save confirmation - Update notification - History display

### HOA-INT-IT-BRD-003: Add Board Member

Priority: High Components: UpdateBoardMembersModal.js

**Test Steps:** 1. Open member management 2. Search for resident 3. Select member role: - Verify eligibility - Check role options - Set permissions 4. Set term dates 5. Confirm addition 6. Check member listing 7. Test access grant 8. Verify notifications

**Expected Results:** - Search works - Role selection functions - Dates validate - Member adds - Listing updates - Access grants - Notifications send - UI refreshes

**Visual Verification Points:** - Search interface - Role selection - Date inputs - Success message - List update

### HOA-INT-IT-BRD-004: Remove Board Member

Priority: High Components: UpdateBoardMembersModal.js

**Test Steps:** 1. Select board member 2. Initiate removal 3. Confirm action 4. Process removal: - End date setting - Permission revocation - Role cleanup 5. Update member list 6. Check access removal 7. Verify notifications 8. Test system impact

**Expected Results:** - Selection works - Confirmation shows - Process completes - Permissions revoke - List updates - Access removes - Notifications send - System stable

**Visual Verification Points:** - Selection interface - Confirmation modal - Progress indicator - Success message - List refresh

### HOA-INT-IT-BRD-005: Issue Violation Process

Priority: High Components: IssueViolationModal.js

**Test Steps:** 1. Open violation form 2. Select violation type: - Check categories - Verify amounts 3. Choose resident 4. Enter details: - Description - Date - Photos 5. Review violation 6. Submit issuance 7. Check notifications 8. Verify record creation

**Expected Results:** - Form opens properly - Types load correctly - Resident selection works - Details save - Review displays - Issuance processes - Notifications send - Records create

**Visual Verification Points:** - Form layout - Selection controls - Photo upload - Review display - Success state

### HOA-INT-IT-BRD-006: Violation Type Management

Priority: High Components: UpdateViolationsModal.js

**Test Steps:** 1. Access violation settings 2. Test type creation: - Add new type - Set fine amount - Define description 3. Modify existing types: - Update amounts - Edit descriptions - Change status 4. Test type removal 5. Verify changes 6. Check active violations 7. Test reporting impact 8. Verify notifications

**Expected Results:** - Settings load - Creation works - Updates save - Removal processes - Changes reflect - Violations update - Reports maintain - Notifications send

**Visual Verification Points:** - Settings interface - Form validation - Update status - Confirmation messages - List updates

### HOA-INT-IT-BRD-007: Assessment Issuance Process

Priority: High Components: IssueAssessmentModal.js

**Test Steps:** 1. Open assessment form 2. Select assessment type: - Regular - Special - One-time 3. Choose recipients: - Individual - Group - All residents 4. Set amounts and dates 5. Review details 6. Process issuance 7. Verify notifications 8. Check payment system

**Expected Results:** - Form displays - Types load - Selection works - Details validate - Review shows - Process completes - Notifications send - Payments enable

**Visual Verification Points:** - Form display - Selection controls - Review screen - Progress indicator - Success message

### HOA-INT-IT-BRD-008: Assessment Rate Management

Priority: High Components: ManageAssessmentsModal.js

**Test Steps:** 1. Access rate management 2. Test rate creation: - Add new rate - Set amounts - Define periods 3. Modify existing rates: - Update amounts - Change periods - Adjust status 4. Review changes 5. Apply updates 6. Check active assessments 7. Verify calculations 8. Test billing impact

**Expected Results:** - Management opens - Creation works - Updates save - Changes apply - Assessments update - Calculations correct - Billing adjusts - System stable

**Visual Verification Points:** - Rate interface - Form validation - Update status - Success messages - List refresh

### HOA-INT-IT-BRD-009: New Account Creation

Priority: High Components: NewAccountModal.js

**Test Steps:** 1. Open account form 2. Enter property details: - Address - Unit number - Property type 3. Set owner information: - Contact details - Initial settings 4. Configure billing: - Start date - Initial charges 5. Review setup 6. Create account 7. Verify notifications 8. Check system access

**Expected Results:** - Form loads - Details save - Settings apply - Creation succeeds - Notifications send - Access generates - History creates - System updates

**Visual Verification Points:** - Form interface - Validation states - Review screen - Progress tracking - Success notice

### HOA-INT-IT-BRD-010: Account Setup Completion

Priority: High Components: NewAccountModal.js

**Test Steps:** 1. Process new account 2. Test welcome setup: - Email generation - Login credentials - Initial password 3. Verify documentation 4. Check system access: - Portal setup - Payment activation - Document access 5. Test notifications 6. Verify permissions 7. Check reporting 8. Monitor first login

**Expected Results:** - Process completes - Welcome sends - Credentials generate - Access grants - Notifications work - Permissions set - Reports update - Login enables

**Visual Verification Points:** - Setup progress - Email preview - Access confirmation - Success status - System updates

### 

## Part 6: Announcement Flow Tests

### HOA-INT-IT-ANC-001: New Announcement Creation

Priority: High Components: CreateAnnouncementModal.js

**Test Steps:** 1. Access announcement creation: - Navigate to Announcements section - Click “Create New Announcement” button 2. Fill creation form: - Enter title - Select type (Event/Announcement) - Enter content - Set dates (if event) - Add location (if event) 3. Test required fields validation 4. Test character limits 5. Preview announcement 6. Submit creation 7. Verify display in list 8. Check notification triggers

**Expected Results:** - Creation modal opens properly - Form validation works - Required fields enforced - Preview displays correctly - Submission successful - List updates immediately - Notifications sent - Proper timestamp shown

**Visual Verification Points:** - Modal rendering - Form field states - Validation messages - Preview formatting - Success confirmation - List update animation

### HOA-INT-IT-ANC-002: Media Attachment Process

Priority: High Components: CreateAnnouncementModal.js

**Test Steps:** 1. Start new announcement 2. Test media upload: - Image selection - File size validation - Format validation - Resolution check 3. Test preview functionality 4. Test multiple uploads 5. Remove media 6. Replace media 7. Complete announcement 8. Verify media display

**Expected Results:** - Upload control works - Size limits enforced - Formats validated - Preview displays - Multiple files handled - Removal works - Replacement successful - Display correct

**Visual Verification Points:** - Upload interface - Preview rendering - Progress indicator - Error messages - Image scaling - Thumbnail display

### HOA-INT-IT-ANC-003: Edit Content Process

Priority: High Components: ManageAnnouncementsModal.js

**Test Steps:** 1. Select existing announcement 2. Access edit mode 3. Modify content: - Title changes - Content updates - Date adjustments - Location updates 4. Test validation rules 5. Save changes 6. Verify updates 7. Check history 8. Test cancellation

**Expected Results:** - Edit mode accessible - Changes save properly - Validation maintains - Updates reflect immediately - History tracks changes - Cancel preserves original - UI updates properly - Notifications handled

**Visual Verification Points:** - Edit mode indication - Form population - Validation states - Update confirmation - List refresh

### HOA-INT-IT-ANC-004: Media Update Process

Priority: High Components: ManageAnnouncementsModal.js

**Test Steps:** 1. Select announcement with media 2. Access media editor 3. Test updates: - Replace image - Add new media - Remove existing - Reorder items 4. Preview changes 5. Save updates 6. Verify display 7. Check mobile view 8. Test cancellation

**Expected Results:** - Media editor opens - Updates process - Preview accurate - Changes save - Display updates - Mobile renders - Cancel works - History maintained

**Visual Verification Points:** - Editor interface - Preview display - Update progress - Success message - Mobile layout

### HOA-INT-IT-ANC-005: Delete Announcement Process

Priority: High Components: ManageAnnouncementsModal.js

**Test Steps:** 1. Select announcement 2. Initiate deletion 3. Confirm deletion 4. Test scenarios: - Future events - Past events - With responses - With media 5. Verify removal 6. Check related items 7. Test cancel option 8. Verify history

**Expected Results:** - Deletion prompt shows - Confirmation required - Removal successful - Related items handled - List updates - History maintained - Cancel works - Notifications sent

**Visual Verification Points:** - Confirmation modal - Progress indicator - Success message - List update - History display

### HOA-INT-IT-ANC-006: List View Process

Priority: High Components: AnnouncementsNews.js

**Test Steps:** 1. Access announcements list 2. Test view options: - Grid view - List view - Calendar view 3. Verify display elements: - Title display - Date formatting - Type indicators - Media thumbnails 4. Test sorting: - By date - By type - By status 5. Test filtering: - By date range - By type - By status 6. Check pagination 7. Verify responsive layout 8. Test refresh functionality

**Expected Results:** - Views switch correctly - Elements display properly - Sorting works - Filters apply correctly - Pagination functions - Layout adapts - Refresh updates - All data accurate

**Visual Verification Points:** - View transitions - Element alignment - Sort indicators - Filter status - Page navigation - Mobile layout

### HOA-INT-IT-ANC-007: Detail View Process

Priority: High Components: AnnouncementModal.js

**Test Steps:** 1. Select announcement 2. Verify detail display: - Full content - Media display - Date information - Location details 3. Test media interaction: - Image enlargement - Gallery navigation 4. Check formatting 5. Test sharing options 6. Verify event details 7. Test mobile view 8. Close and return

**Expected Results:** - Details load properly - Media displays correctly - Dates formatted - Location shown - Media controls work - Sharing functions - Mobile renders - Navigation works

**Visual Verification Points:** - Content layout - Media sizing - Date format - Location display - Share options - Mobile display

### HOA-INT-IT-ANC-008: Calendar Integration Process

Priority: High Components: AnnouncementCalendar.js

**Test Steps:** 1. Access calendar view 2. Verify event display: - Date accuracy - Time slots - Event indicators 3. Test navigation: - Month view - Week view - Day view 4. Check event details: - Click events - Hover info - Duration display 5. Test filters 6. Verify updates 7. Check mobile view 8. Test event selection

**Expected Results:** - Calendar loads - Events show correctly - Navigation works - Details display - Filters function - Updates reflect - Mobile renders - Selection works

**Visual Verification Points:** - Calendar grid - Event markers - Time indicators - Detail popups - Mobile layout - Selection state

### HOA-INT-IT-ANC-009: Calendar Updates Process

Priority: High Components: AnnouncementCalendar.js

**Test Steps:** 1. Create new event 2. Verify calendar update: - Immediate display - Correct placement - Visual indicator 3. Modify event: - Date change - Time change - Duration change 4. Delete event 5. Test recurring events 6. Check conflict handling 7. Verify notifications 8. Test sync status

**Expected Results:** - Updates show immediately - Placement accurate - Changes reflect - Deletion removes - Recurrence works - Conflicts handled - Notifications sent - Sync maintains

**Visual Verification Points:** - Update animation - Event position - Change indicators - Removal effect - Conflict warning - Sync status

### HOA-INT-IT-ANC-010: Content Search Process

Priority: High Components: AnnouncementsNews.js

**Test Steps:** 1. Access search function 2. Test search criteria: - Title search - Content search - Date range - Type filter 3. Test search execution 4. Verify results: - Result accuracy - Result order - Result count 5. Test empty results 6. Check highlight function 7. Test clear search 8. Verify performance

**Expected Results:** - Search activates - Criteria applies - Results display - Ordering correct - Count accurate - Empty handled - Highlights show - Performance good

**Visual Verification Points:** - Search interface - Result display - Highlight effect - Empty state - Loading indicator - Clear function

## 

## Part 7: Survey Flow Tests

### HOA-INT-IT-SRV-001: New Survey Creation Process

Priority: High Components: CreateSurveyModal.js

**Test Steps:** 1. Access survey creation: - Navigate to Surveys - Click “Create Survey” button 2. Enter survey details: - Title/question input - End date selection - Response options (2-4) 3. Test validation: - Required fields - Date validation - Option limits 4. Preview survey 5. Submit creation 6. Verify survey listing 7. Check notification delivery 8. Test mobile creation

**Expected Results:** - Creation modal opens - Form validation works - Date picker functions - Options add/remove - Preview displays - Submission succeeds - List updates - Notifications send

**Visual Verification Points:** - Modal display - Form validation states - Preview rendering - Success confirmation - List update animation - Mobile responsiveness

### HOA-INT-IT-SRV-002: Question Types Process

Priority: High Components: CreateSurveyModal.js

**Test Steps:** 1. Test multiple answer formats: - Two options - Three options - Four options 2. Verify option handling: - Add options - Remove options - Edit options - Reorder options 3. Test character limits 4. Check formatting 5. Preview display 6. Save changes 7. View on dashboard 8. Test mobile display

**Expected Results:** - All formats work - Options manage correctly - Limits enforce - Format maintains - Preview accurate - Changes save - Display correct - Mobile renders

**Visual Verification Points:** - Option controls - Format preview - Validation messages - Display layout - Mobile view - Dashboard integration

### HOA-INT-IT-SRV-003: Submit Response Process

Priority: High Components: SurveyCompletionModal.js

**Test Steps:** 1. Access active survey 2. Test response process: - Option selection - Selection change - Submit action 3. Verify validation: - Required response - Single selection 4. Check submission: - Processing indicator - Success confirmation - Result update 5. Test multiple responses 6. Verify response record 7. Check mobile submission 8. Test offline handling

**Expected Results:** - Survey accessible - Selection works - Validation functions - Submission processes - Results update - Records maintain - Mobile works - Offline handles

**Visual Verification Points:** - Selection interface - Submit button state - Processing indicator - Success message - Results display - Mobile layout

### HOA-INT-IT-SRV-004: Response Modification Process

Priority: High Components: SurveyCompletionModal.js

**Test Steps:** 1. Locate existing response 2. Test modification: - Change selection - Update submission - Cancel change 3. Verify validation 4. Check update process 5. Test time constraints 6. Verify result updates 7. Check history 8. Test mobile update

**Expected Results:** - Response editable - Changes save - Validation works - Updates reflect - Time rules apply - Results update - History tracks - Mobile functions

**Visual Verification Points:** - Edit interface - Update indicators - Time warnings - Success message - History display - Mobile layout

### HOA-INT-IT-SRV-005: Results Visualization Process

Priority: High Components: SurveyResultsModal.js

**Test Steps:** 1. Access survey results 2. Test display options: - Chart view - Table view - Percentage view 3. Verify calculations 4. Test filtering options 5. Check data accuracy 6. Test export options 7. Verify real-time updates 8. Test mobile display

**Expected Results:** - Results load - Views switch - Calculations correct - Filters work - Data accurate - Export functions - Updates show - Mobile displays

**Visual Verification Points:** - Chart rendering - Table formatting - Update animations - Export options - Filter controls - Mobile layout

### HOA-INT-IT-SRV-006: Results Export Process

Priority: High Components: SurveyResultsModal.js

**Test Steps:** 1. Access survey results 2. Test export formats: - PDF generation - CSV download - Data formatting 3. Verify export content: - Question details - Response data - Charts/graphs - Timestamps 4. Test batch export 5. Check file naming 6. Verify data accuracy 7. Test download process 8. Validate file formats

**Expected Results:** - Exports generate - Formats correct - Content complete - Batch works - Names proper - Data accurate - Downloads work - Files valid

**Visual Verification Points:** - Export controls - Progress indicator - Success message - File names - Format options - Download status

### HOA-INT-IT-SRV-007: Active Surveys Display

Priority: High Components: Survey.js

**Test Steps:** 1. View active surveys 2. Verify display elements: - Survey titles - End dates - Response counts - Status indicators 3. Test sorting options: - By date - By responses - By title 4. Check filtering 5. Verify accessibility 6. Test refresh function 7. Check mobile view 8. Verify user access

**Expected Results:** - List displays - Elements show - Sorting works - Filters apply - Access controls - Refresh updates - Mobile renders - Access proper

**Visual Verification Points:** - List layout - Sort indicators - Filter status - Access levels - Mobile display - Update animation

### HOA-INT-IT-SRV-008: Past Surveys History

Priority: High Components: Survey.js

**Test Steps:** 1. Access survey history 2. Verify archived surveys: - Completed surveys - Results display - Date information - Participation data 3. Test filtering: - Date range - Response rate - Category 4. Check search function 5. Verify data accuracy 6. Test export options 7. Check mobile access 8. Verify permissions

**Expected Results:** - History loads - Archives show - Filters work - Search functions - Data accurate - Export works - Mobile displays - Permissions apply

**Visual Verification Points:** - Archive layout - Filter controls - Search interface - Export options - Mobile view - Permission indicators

### HOA-INT-IT-SRV-009: Survey Removal Process

Priority: High Components: Survey.js

**Test Steps:** 1. Select survey for removal 2. Test deletion process: - Active surveys - Completed surveys - With responses 3. Verify confirmation: - Warning message - Data implications - Cancellation option 4. Check removal impact: - List update - Data cleanup - User notifications 5. Test batch deletion 6. Verify audit trail 7. Check mobile deletion 8. Test permission levels

**Expected Results:** - Selection works - Process confirms - Warnings show - Removal complete - Lists update - Data cleans - Audit records - Permissions work

**Visual Verification Points:** - Selection interface - Warning modal - Progress indicator - Success message - List updates - Mobile interface

### HOA-INT-IT-SRV-010: Survey Access Control

Priority: High Components: Survey.js

**Test Steps:** 1. Test access levels: - Admin access - Board access - Resident access 2. Verify permissions: - Create rights - Edit rights - Delete rights - View rights 3. Test restriction enforcement 4. Check time-based access 5. Verify user notification 6. Test invalid access 7. Check mobile permissions 8. Verify audit logging

**Expected Results:** - Levels enforce - Permissions work - Restrictions apply - Timing controls - Notifications send - Invalid blocks - Mobile works - Logging records

**Visual Verification Points:** - Permission controls - Access indicators - Error messages - Time restrictions - Mobile access - Audit display

### 

## Part 8: Profile Management Flow Tests

### HOA-INT-IT-PRF-001: Personal Info Update Process

Priority: High Components: PersonalInfoModal.js

**Test Steps:** 1. Access personal info section 2. Test field updates: - First name - Last name - Other personal fields 3. Verify validation: - Required fields - Format rules - Character limits 4. Submit changes 5. Verify data persistence 6. Check display updates 7. Test cancel function 8. Verify audit logging

**Expected Results:** - Edit form opens - Fields update - Validation works - Changes save - Display updates - Cancel preserves - Audit logs - UI refreshes

**Visual Verification Points:** - Form display - Validation messages - Success confirmation - Updated content - Mobile layout

### HOA-INT-IT-PRF-002: Contact Info Update Process

Priority: High Components: ContactInfoModal.js

**Test Steps:** 1. Access contact info 2. Test updates: - Email address - Phone number - Address details 3. Verify validation: - Email format - Phone format - Required fields 4. Submit changes 5. Test verification processes 6. Check notification delivery 7. Verify system updates 8. Test mobile updates

**Expected Results:** - Edit form loads - Updates process - Validation runs - Changes save - Verification works - Notifications send - System updates - Mobile functions

**Visual Verification Points:** - Form layout - Error messages - Success states - Update display - Mobile view

### HOA-INT-IT-PRF-003: Notification Preferences Process

Priority: High Components: NotificationPreferencesModal.js

**Test Steps:** 1. Access preferences 2. Test toggles: - Email notifications - News updates - Payment alerts - Document notices 3. Test bulk actions 4. Save preferences 5. Verify settings 6. Check notification test 7. Test mobile interface 8. Verify persistence

**Expected Results:** - Preferences load - Toggles work - Bulk actions function - Settings save - Tests send - Mobile works - Settings persist - UI updates

**Visual Verification Points:** - Toggle states - Save confirmation - Test results - Mobile layout - Setting indicators

### HOA-INT-IT-PRF-004: Password Change Process

Priority: High Components: ChangePasswordModal.js

**Test Steps:** 1. Access password change 2. Test validation: - Current password - New password rules - Confirmation match 3. Submit change 4. Verify session handling 5. Test login with new password 6. Check notification 7. Verify security log 8. Test mobile change

**Expected Results:** - Form displays - Validation works - Change processes - Session maintains - Login works - Notifications send - Logs update - Mobile functions

**Visual Verification Points:** - Form display - Validation states - Success message - Session status - Mobile layout

### HOA-INT-IT-PRF-005: Email Verification Process

Priority: High Components: OwnerInfo.js

**Test Steps:** 1. Initiate verification 2. Test email delivery 3. Verify link function 4. Check token validation 5. Test expiration 6. Verify status update 7. Check mobile process 8. Test resend function

**Expected Results:** - Process starts - Email sends - Link works - Token validates - Expiration handles - Status updates - Mobile works - Resend functions

**Visual Verification Points:** - Status indicators - Verification UI - Success message - Email format - Mobile display

### HOA-INT-IT-PRF-006: Profile View Load Process

Priority: High Components: OwnerInfo.js

**Test Steps:** 1. Access profile view 2. Verify data display: - Personal info section - Contact details - Account status - Preferences 3. Test section navigation 4. Check data accuracy 5. Verify permissions 6. Test refresh function 7. Check mobile view 8. Verify data protection

**Expected Results:** - Profile loads - All sections show - Navigation works - Data accurate - Permissions apply - Refresh updates - Mobile displays - Protection works

**Visual Verification Points:** - Section layout - Data formatting - Navigation controls - Mobile responsiveness - Permission indicators

### HOA-INT-IT-PRF-007: Settings Sync Process

Priority: High Components: Settings.js

**Test Steps:** 1. Modify settings 2. Test sync process: - Local storage - Server sync - Conflict handling 3. Verify across devices 4. Check offline handling 5. Test sync indicators 6. Verify error handling 7. Check mobile sync 8. Test recovery process

**Expected Results:** - Changes save - Sync executes - Conflicts resolve - Devices update - Offline works - Indicators show - Errors handle - Recovery works

**Visual Verification Points:** - Sync status - Error messages - Success indicators - Mobile states - Recovery options

### HOA-INT-IT-PRF-008: Theme Toggle Process

Priority: High Components: Settings.js

**Test Steps:** 1. Access theme settings 2. Test toggle function: - Light mode - Dark mode - System preference 3. Verify immediate update 4. Check persistence 5. Test component updates 6. Verify preference save 7. Check mobile display 8. Test system changes

**Expected Results:** - Settings access - Toggle works - Updates apply - Settings persist - Components update - Preferences save - Mobile renders - System adapts

**Visual Verification Points:** - Theme switch - Color changes - Animation effects - Mobile appearance - System integration

### HOA-INT-IT-PRF-009: Data Export Process

Priority: High Components: OwnerInfo.js

**Test Steps:** 1. Access export function 2. Test export options: - Profile data - Activity history - Settings data 3. Verify file generation 4. Check data format 5. Test download process 6. Verify data accuracy 7. Check mobile export 8. Test format options

**Expected Results:** - Export initiates - Options work - Files generate - Format correct - Downloads work - Data accurate - Mobile functions - Formats available

**Visual Verification Points:** - Export controls - Progress indicator - Download status - Format selection - Mobile interface

### HOA-INT-IT-PRF-010: Account Status Process

Priority: High Components: OwnerInfo.js

**Test Steps:** 1. View account status 2. Verify status indicators: - Account standing - Payment status - Violations status - Access level 3. Test status updates 4. Check notification system 5. Verify history log 6. Test mobile display 7. Check real-time updates 8. Verify permission levels

**Expected Results:** - Status displays - Indicators work - Updates process - Notifications send - History logs - Mobile shows - Updates real-time - Permissions work

**Visual Verification Points:** - Status display - Indicator colors - Update animations - Mobile layout - Permission controls

### 

## Part 9: Assessment Flow Tests

### HOA-INT-IT-ASS-001: New Assessment Type Creation

Priority: High Components: ManageAssessmentsModal.js, apiService.js

**Test Steps:** 1. Access assessment management 2. Select “Add New Type” 3. Enter type details: - Description - Category - Fee structure 4. Save new type 5. Verify type listing 6. Test duplicate prevention 7. Check validation rules

**Expected Results:** - Type form displays correctly - Validation functions properly - Type saves successfully - List updates immediately - Duplicates prevented - Rules enforced - UI updates accordingly

**Visual Verification Points:** - Form rendering - Validation messages - Success confirmation - Updated type list - Error handling display

### HOA-INT-IT-ASS-002: Assessment Rate Creation

Priority: High Components: ManageAssessmentsModal.js, apiService.js

**Test Steps:** 1. Access rate management 2. Add new rate: - Amount entry - Year specification - Type association 3. Test validation: - Minimum amount - Maximum amount - Year format 4. Save rate 5. Verify rate listing 6. Check type association 7. Test rate availability

**Expected Results:** - Rate form opens properly - Validation works correctly - Rate saves successfully - List updates properly - Type links correctly - Rate available for use - Proper error handling

**Visual Verification Points:** - Form display - Validation states - Success message - Rate list update - Type association display

### HOA-INT-IT-ASS-003: Bulk Assessment Issuance

Priority: High Components: IssueAssessmentModal.js, apiService.js

**Test Steps:** 1. Navigate to issue assessments 2. Select multiple residents 3. Choose assessment type 4. Set parameters: - Due date - Amount per unit - Late fee rules 5. Review selection 6. Process bulk issuance 7. Monitor progress 8. Verify notifications 9. Check account updates

**Expected Results:** - Multiple selection works - Parameters apply to all - Processing shows progress - All assessments created - Notifications sent - Accounts updated - History records created

**Visual Verification Points:** - Selection interface - Progress indicator - Confirmation message - Batch status display - Account updates - History entries

### HOA-INT-IT-ASS-004: Single Assessment Issuance

Priority: High Components: IssueAssessmentModal.js, apiService.js

**Test Steps:** 1. Select single resident 2. Choose assessment type 3. Enter assessment details: - Amount - Due date - Description 4. Review details 5. Submit assessment 6. Check notification 7. Verify account 8. Test immediate access

**Expected Results:** - Single selection works - Details save correctly - Assessment creates - Notification sends - Account updates - History records - Immediate visibility

**Visual Verification Points:** - Selection interface - Form validation - Success message - Account display - History entry

### HOA-INT-IT-ASS-005: Yearly Assessment Setup

Priority: High Components: ManageAssessmentsModal.js, apiService.js

**Test Steps:** 1. Access yearly settings 2. Configure parameters: - Annual rate - Due dates - Payment schedule 3. Set notifications 4. Review configuration 5. Save settings 6. Test automatic creation 7. Verify scheduling

**Expected Results:** - Configuration saves - Schedule sets correctly - Notifications configure - Automatic creation works - Schedule maintained - System validates rules - Settings persist

**Visual Verification Points:** - Configuration form - Schedule display - Validation feedback - Success confirmation - Schedule review

### HOA-INT-IT-ASS-006: Assessment Status Updates

Priority: High Components: IssueAssessmentModal.js, apiService.js

**Test Steps:** 1. View assessment list 2. Select assessment 3. Change status: - Pending to Active - Active to Paid - Active to Past Due 4. Update comments 5. Save changes 6. Check history 7. Verify notifications

**Expected Results:** - Status changes save - Comments update - History records - Notifications send - Account reflects change - UI updates - Changes persist

**Visual Verification Points:** - Status indicators - Update interface - History display - Notification preview - Account status

### HOA-INT-IT-ASS-007: Email Notification System

Priority: High Components: IssueAssessmentModal.js, apiService.js

**Test Steps:** 1. Trigger notifications: - New assessment - Due date reminder - Past due notice 2. Check email format 3. Verify content 4. Test links 5. Monitor delivery 6. Check bounce handling 7. Verify tracking

**Expected Results:** - Emails trigger correctly - Format is correct - Content accurate - Links functional - Delivery tracked - Bounces handled - System updated

**Visual Verification Points:** - Email preview - Tracking interface - Delivery status - Bounce notifications - System logs

### HOA-INT-IT-ASS-008: Permission-Based Access

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Test different roles: - Admin access - Board member access - Regular user access 2. Check feature visibility 3. Test action permissions 4. Verify restrictions 5. Monitor access logs 6. Test override attempts 7. Check error handling

**Expected Results:** - Roles enforce correctly - Features show/hide properly - Actions restricted - Attempts logged - Errors handled - Override prevented - Access maintained

**Visual Verification Points:** - Role-based UI - Permission messages - Error displays - Access indicators - Log entries

### HOA-INT-IT-ASS-009: Error Handling and Display

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Test error scenarios: - Invalid amounts - Missing data - System errors - Network issues 2. Monitor error messages 3. Check recovery options 4. Test data persistence 5. Verify state management 6. Check error logging 7. Test user feedback

**Expected Results:** - Errors caught properly - Messages display clearly - Recovery options shown - Data preserves - State maintains - Logs record errors - Feedback appropriate

**Visual Verification Points:** - Error messages - Recovery UI - State preservation - Log entries - User guidance

### HOA-INT-IT-ASS-010: Data Refresh and Updates

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Monitor data updates 2. Test refresh triggers: - Manual refresh - Auto refresh - Event-based refresh 3. Check update frequency 4. Verify data accuracy 5. Test concurrent updates 6. Monitor performance 7. Check UI updates

**Expected Results:** - Updates trigger properly - Data refreshes correctly - Frequency maintained - Accuracy preserved - Concurrency handled - Performance stable - UI updates smoothly

**Visual Verification Points:** - Refresh indicators - Update animations - Loading states - Data display - UI transitions

### 

## Part 10: Violation Flow Tests

### HOA-INT-IT-VIO-001: New Violation Type Creation

Priority: High Components: UpdateViolationsModal.js, apiService.js

**Test Steps:** 1. Access violation types 2. Create new type: - Description - Fine amount - Category 3. Set validation rules 4. Save new type 5. Verify type listing 6. Test duplicate check 7. Check availability

**Expected Results:** - Type creation works - Validation functions - Rules save properly - List updates - Duplicates prevented - Type available - UI refreshes

**Visual Verification Points:** - Creation form - Validation feedback - Success message - Updated list - Error handling

### HOA-INT-IT-VIO-002: Violation Type Update

Priority: High Components: UpdateViolationsModal.js, apiService.js

**Test Steps:** 1. Select existing type 2. Modify details: - Description update - Amount change - Rule adjustments 3. Save changes 4. Verify updates 5. Check existing violations 6. Test rate applications 7. Monitor history

**Expected Results:** - Updates save properly - Changes reflect - Existing violations maintained - Rates apply correctly - History records - UI updates - Changes persist

**Visual Verification Points:** - Edit interface - Update confirmation - History display - Rate application - UI refresh

### HOA-INT-IT-VIO-003: Single Violation Issuance

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Select resident 2. Choose violation type 3. Enter details: - Date of violation - Description - Location 4. Add supporting notes 5. Review violation 6. Submit issuance 7. Verify notification 8. Check resident record

**Expected Results:** - Selection works - Details save correctly - Notes included - Issuance processes - Notification sends - Record updates - History creates - UI refreshes

**Visual Verification Points:** - Selection interface - Form validation - Success message - Record display - History update

### HOA-INT-IT-VIO-004: Media Upload Handling

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Select violation 2. Upload media: - Photos - Documents - Multiple files 3. Test file types 4. Check size limits 5. Verify storage 6. Test retrieval 7. Check display

**Expected Results:** - Upload works - Types validated - Size checked - Storage successful - Retrieval functions - Display correct - Links work

**Visual Verification Points:** - Upload interface - Progress indicator - Preview display - Error messages - File listing

### HOA-INT-IT-VIO-005: Notice Generation

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Trigger notice creation 2. Check template: - Violation details - Fine amount - Due date 3. Generate notice 4. Verify format 5. Test delivery 6. Check tracking 7. Monitor status

**Expected Results:** - Notice generates - Template correct - Details included - Format proper - Delivery works - Tracking functions - Status updates

**Visual Verification Points:** - Notice preview - Format check - Delivery status - Tracking display - Status indicators

### HOA-INT-IT-VIO-006: Payment Creation

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Process violation fine 2. Create payment: - Amount calculation - Due date setting - Payment options 3. Link to violation 4. Test payment flow 5. Verify records 6. Check reconciliation 7. Monitor status

**Expected Results:** - Payment creates - Amount correct - Links properly - Flow works - Records update - Reconciles - Status tracks

**Visual Verification Points:** - Payment form - Link display - Status updates - Record view - Reconciliation

### HOA-INT-IT-VIO-007: Permission Check Process

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Test user roles: - Admin access - Board member levels - Regular users 2. Verify permissions: - Create violations - Edit violations - Delete violations 3. Check override attempts 4. Test delegation rules 5. Monitor access logs 6. Verify restrictions 7. Test error messages

**Expected Results:** - Roles enforce correctly - Permissions work - Overrides blocked - Rules followed - Logs record - Restrictions hold - Errors display

**Visual Verification Points:** - Access controls - Permission errors - Log entries - Restriction notices - Error displays

### HOA-INT-IT-VIO-008: Error Handling

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Test error cases: - Invalid data entry - System errors - Network issues - Validation failures 2. Check error messages 3. Test recovery options 4. Verify data integrity 5. Monitor error logs 6. Test user feedback 7. Check state preservation

**Expected Results:** - Errors caught - Messages clear - Recovery works - Data preserved - Logs complete - Feedback appropriate - State maintains

**Visual Verification Points:** - Error messages - Recovery options - Log displays - Feedback forms - State indicators

### HOA-INT-IT-VIO-009: History Record Creation

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Create violation 2. Check history entry: - Timestamp - Details - User info - Changes 3. Test modifications 4. Verify audit trail 5. Check access control 6. Test filtering 7. Verify retention

**Expected Results:** - History creates - Details complete - Changes track - Trail maintains - Access controls - Filters work - Retention proper

**Visual Verification Points:** - History display - Detail view - Filter interface - Access controls - Retention info

### HOA-INT-IT-VIO-010: Multiple Violation Processing

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Create multiple violations: - Same resident - Different types - Different dates 2. Test batch processing 3. Check notifications 4. Verify records 5. Test payment handling 6. Monitor status updates 7. Check history

**Expected Results:** - Batch works - Records create - Notifications send - Payments link - Status updates - History records - System performs

**Visual Verification Points:** - Batch interface - Status displays - Record views - Payment links - History entries

### HOA-INT-IT-VIO-011: Due Date Handling

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Set violation due dates: - Initial notice - Grace period - Final due date 2. Test date calculations 3. Check notifications 4. Monitor escalations 5. Verify reminders 6. Test overdue handling 7. Check payment windows

**Expected Results:** - Dates set correctly - Calculations accurate - Notifications trigger - Escalations work - Reminders send - Overdue handled - Windows enforce

**Visual Verification Points:** - Date displays - Notification timing - Escalation indicators - Reminder notices - Status updates

### HOA-INT-IT-VIO-012: Status Update Management

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Update violation status: - New to Active - Active to Resolved - Active to Escalated 2. Add status notes 3. Test state changes 4. Verify notifications 5. Check history 6. Monitor triggers 7. Test status rules

**Expected Results:** - Status updates work - Notes save - Changes record - Notifications send - History updates - Triggers function - Rules enforce

**Visual Verification Points:** - Status indicators - Note display - History view - Notification preview - Rule enforcement

### HOA-INT-IT-VIO-013: Data Refresh and Updates

Priority: High Components: Both Modals, apiService.js

**Test Steps:** 1. Monitor real-time updates: - Status changes - Payment updates - Resolution updates 2. Test refresh triggers: - Automatic refresh - Manual refresh - Event-based refresh 3. Verify data consistency 4. Check update frequency 5. Test concurrent updates 6. Monitor performance 7. Verify UI updates

**Expected Results:** - Updates occur in real-time - Refresh functions work - Data remains consistent - Frequency appropriate - Concurrency handled - Performance maintained - UI updates smoothly

**Visual Verification Points:** - Update indicators - Refresh buttons - Loading states - Data consistency - UI responsiveness

### HOA-INT-IT-VIO-014: Resident Search Integration

Priority: High Components: IssueViolationModal.js, apiService.js

**Test Steps:** 1. Test search functionality: - Name search - Address search - Account number search 2. Verify search results 3. Test filtering options 4. Check result accuracy 5. Monitor search performance 6. Test partial matches 7. Verify result selection

**Expected Results:** - Search works correctly - Results accurate - Filters function - Performance good - Matches found - Selection works - UI responsive

**Visual Verification Points:** - Search interface - Results display - Filter controls - Selection highlight - Performance indicators

### HOA-INT-IT-VIO-015: Dashboard Updates

Priority: High Components: Both Modals, Dashboard Components

**Test Steps:** 1. Create new violation 2. Check dashboard updates: - Violation count - Status summaries - Recent activities 3. Test filter impacts 4. Verify metrics 5. Check chart updates 6. Monitor refresh timing 7. Test interactive elements

**Expected Results:** - Dashboard updates - Counts accurate - Summaries reflect changes - Filters work - Metrics calculate - Charts refresh - Interaction works

**Visual Verification Points:** - Dashboard display - Counter updates - Chart refreshes - Filter effects - Interactive responses

## Test Status Matrix

| Test ID | Status | Last Executed | Pass/Fail | Notes |
| --- | --- | --- | --- | --- |
| HOA-INT-IT-AUTH-001 through 010 | Testing Complete | 11/30 | Pass | Authentication Tests |
| HOA-INT-IT-PAY-001 through 010 | Testing Complete | 11/30 | Pass | Payment Flow Tests |
| HOA-INT-IT-ASS-001 through 010 | Testing Complete | 11/30 | Pass | Assessment Flow Tests |
| HOA-INT-IT-VIO-001 through 015 | Testing Complete | 11/30 | Pass | Violation Flow Tests |
| HOA-INT-IT-DOC-001 through 010 | Testing Complete | 11/30 | Pass | Document Flow Tests |
| HOA-INT-IT-MSG-001 through 010 | Testing Complete | 11/30 | Pass | Message Flow Tests |
| HOA-INT-IT-BRD-001 through 010 | Testing Complete | 11/30 | Pass | Board Member Flow Tests |
| HOA-INT-IT-ANC-001 through 010 | Testing Complete | 11/30 | Pass | Announcement Flow Tests |
| HOA-INT-IT-SRV-001 through 010 | Testing Complete | 11/30 | Pass | Survey Flow Tests |
| HOA-INT-IT-PRF-001 through 010 | Testing Complete | 11/30 | Pass | Profile Management Tests |

## Execution Timeline

* Start Date: November 30, 2024
* Authentication Flow Tests: November 30
* Payment Flow Tests: November 30
* Assessment Flow Tests: November 30
* Violation Flow Tests: November 30
* Document Flow Tests: November 30
* Message Flow Tests: November 30
* Board Member Flow Tests: November 30
* Announcement Flow Tests: November 30
* Survey Flow Tests: November 30
* Profile Management Tests: November 30
* Documentation & Review: November 30
* Completion Date: November 30, 2024

## Approval Section

| Role | Name | Date | Signature |
| --- | --- | --- | --- |
| Developer/Tester | Justin Pike | \_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_ |