

Notification System Testing Report

Generated: December 3, 2025

Status: Test Suite Created - Ready for Execution

Executive Summary

A comprehensive test suite has been created for the Mindful Champion notification system. The suite consists of **7 test modules** covering all critical functionality from API endpoints to end-to-end user workflows.

Test Suite Overview

Test Structure

```
scripts/tests/notification-system/  
├── preferences-tests.ts      # Notification preferences API tests  
├── reminders-tests.ts       # Reminders dashboard CRUD tests  
├── coach-kai-tests.ts       # Coach Kai integration tests  
├── goal-notifications-tests.ts # Goal notification flow tests  
├── email-tests.ts           # Email delivery tests  
├── cron-tests.ts            # Cron job processing tests  
├── integration-tests.ts     # End-to-end integration tests  
└── run-all-tests.ts        # Main test runner
```

Test Modules

1. Notification Preferences Tests (preferences-tests.ts)

Coverage: API endpoints for user notification preferences

Tests:

- ✓ GET /api/notifications/preferences - Fetch user preferences
- ✓ PUT /api/notifications/preferences - Update preferences
- ✓ Timezone handling in preferences
- ✓ Error handling for invalid categories
- ✓ Database update verification
- ✓ All notification categories present (GOALS, VIDEO_ANALYSIS, TOURNAMENTS, etc.)

Key Validations:

- Preference structure integrity
- All 7 categories properly initialized
- Timezone conversion accuracy
- Database persistence

2. Reminders Dashboard Tests (reminders-tests.ts)

Coverage: CRUD operations for user reminders

Tests:

- ☒ POST /api/notifications/reminders - Create reminder
- ☒ GET /api/notifications/reminders - Fetch reminders list
- ☒ PUT /api/notifications/reminders/[id] - Update reminder
- ☒ GET /api/notifications/scheduled - Get upcoming reminders
- ☒ GET /api/notifications/history - Fetch notification history
- ☒ DELETE /api/notifications/reminders/[id] - Delete reminder

Key Validations:

- Reminder creation with proper scheduling
 - Update operations modify correct fields
 - Deletion cascades properly
 - History tracking works correctly
-

3. Coach Kai Integration Tests (coach-kai-tests.ts)

Coverage: Natural language reminder parsing through Coach Kai

Tests:

- ☒ Parse: "Remind me to practice serves tomorrow at 3 PM"
- ☒ Parse: "Set a daily reminder at 8 AM"
- ☒ Parse: "I want to be notified every Monday"
- ☒ Parse: "Daily motivation at 7 AM please"
- ☒ Confirmation message in Coach Kai response
- ☒ Handle ambiguous time expressions
- ☒ Response includes link to reminders dashboard

Key Validations:

- Natural language understanding accuracy
 - Proper time parsing and scheduling
 - Category assignment logic
 - User confirmation flow
-

4. Goal Notifications Tests (goal-notifications-tests.ts)

Coverage: Complete goal lifecycle notification flow

Tests:

- ☒ Create goal with notifications enabled
- ☒ Immediate confirmation email sent
- ☒ Daily reminder properly configured
- ☒ Tips generated for all categories (technique, strategy, mental, physical, practice)
- ☒ Milestone achievement email (50% progress)
- ☒ Goal completion flow
- ☒ Cancel notifications when goal deleted

Key Validations:

- Notification scheduling on goal creation
- Confirmation email delivery

- Daily tips generation system
 - Milestone detection and celebration
 - Cleanup on goal deletion
-

5. Email Delivery Tests (`email-tests.ts`)

Coverage: Gmail SMTP integration and email delivery

Tests:

- ☒ Gmail SMTP connection test
- ☒ Email templates validation
- ☒ Send test emails from all configured addresses
- `NOTIFICATION_EMAIL` (welcomefrommc@mindfulchampion.com)
- `SUPPORT_EMAIL`
- `PARTNERS_EMAIL`
- `SPONSORS_EMAIL`
- ☒ HTML and plain text email formatting
- ☒ Verify email delivery

Key Validations:

- SMTP authentication
 - Template file existence
 - Multi-address support
 - HTML rendering
 - Delivery confirmation
-

6. Cron Job Tests (`cron-tests.ts`)

Coverage: Scheduled notification processing

Tests:

- ☒ `POST /api/notifications/process-pending` - Process due notifications
- ☒ `POST /api/notifications/send-daily-digest` - Send daily summary
- ☒ Cron endpoints require authentication
- ☒ Cron endpoints reject invalid token
- ☒ Scheduled notification processing
- ☒ Failed notifications are retried
- ☒ Max retry limit respected (3 attempts)

Key Validations:

- Authentication security
 - Notification processing logic
 - Retry mechanism
 - Max retry enforcement
 - Status updates
-

7. Integration Tests (integration-tests.ts)

Coverage: End-to-end user workflows

Tests:

- ✓ E2E: New user onboarding with notifications
- ✓ E2E: Multiple goals with different frequencies
- ✓ E2E: Update preferences and verify changes
- ✓ E2E: Ask Coach Kai to set reminder
- ✓ E2E: Notification history is properly tracked

Key Validations:

- Complete user journey
- Multiple notification frequencies
- Cross-component integration
- Data consistency
- History tracking

Test Execution Guide

Prerequisites

1. Database Connection

```
bash
# Ensure database is running and accessible
DATABASE_URL="your-database-url"
```

2. Email Configuration

```
bash
# Gmail SMTP credentials
GMAIL_USER="your-email@gmail.com"
GMAIL_APP_PASSWORD="your-app-password"
```

3. Application Running

```
bash
# Start development server
npm run dev
# OR production server
npm run build && npm start
```

4. Test User

```
bash
# Create test user in database
# Email should contain 'test' for test identification
```

Running Tests

Run All Tests

```
cd scripts/tests/notification-system
npx ts-node run-all-tests.ts
```

Run Individual Test Suites

```
# Preferences tests
npx ts-node preferences-tests.ts

# Reminders tests
npx ts-node reminders-tests.ts

# Coach Kai tests
npx ts-node coach-kai-tests.ts

# Goal notifications tests
npx ts-node goal-notifications-tests.ts

# Email tests
npx ts-node email-tests.ts

# Cron tests
npx ts-node cron-tests.ts

# Integration tests
npx ts-node integration-tests.ts
```

Current Status

✓ Completed

- [x] Test suite structure created
- [x] All 7 test modules implemented
- [x] Test runner with colored output
- [x] Comprehensive test coverage
- [x] Documentation created

⚠ Issues Identified

1. Gmail Authentication

- **Issue:** Gmail SMTP credentials not working
- **Error:** “535-5.7.8 Username and Password not accepted”
- **Solution Required:**
 - Enable 2-factor authentication on Gmail account
 - Generate App Password specifically for SMTP
 - Update `MAIL_APP_PASSWORD` in `.env`

2. Environment Variables

- **Missing:** Some email addresses not fully configured
- **Required:**

```
bash
```

```
NOTIFICATION_EMAIL=welcomefrommc@mindfulchampion.com
```

```
SUPPORT_EMAIL=support@mindfulchampion.com
```

```
PARTNERS_EMAIL=partners@mindfulchampion.com
```

```
SPONSORS_EMAIL=sponsors@mindfulchampion.com
```

3. Authentication in Tests

- **Current:** Using placeholder tokens
- **Recommended:** Implement proper test authentication
- **Alternative:** Use mocked auth for unit tests



Pending Actions

1. Fix Gmail Credentials

- ☐ Enable 2FA on Gmail account
- ☐ Generate App Password
- ☐ Update `.env` file
- ☐ Test email delivery

2. Execute Test Suite

- ☐ Run complete test suite
- ☐ Document actual results
- ☐ Fix any failing tests
- ☐ Verify all integrations

3. Continuous Integration

- ☐ Add test script to `package.json`
 - ☐ Set up automated testing
 - ☐ Create CI/CD pipeline
 - ☐ Add pre-commit hooks
-

Test Coverage Summary

Component	Tests	Coverage
Notification Preferences	5	API endpoints, database, validation
Reminders Dashboard	6	CRUD operations, scheduling, history
Coach Kai Integration	7	NLP parsing, confirmation, links
Goal Notifications	7	Lifecycle, emails, tips, milestones
Email Delivery	5	SMTP, templates, multi-address
Cron Jobs	7	Processing, auth, retry logic
Integration	5	E2E workflows, data consistency
TOTAL	42	Comprehensive

Recommendations

Immediate Actions

- Fix Email Authentication**

```
```bash
Steps:
```

  - Go to Google Account > Security
  - Enable 2-Step Verification
  - Generate App Password for "Mail"
  - Update GMAIL\_APP\_PASSWORD in .env
  - Restart application

```
```
```
- Run Test Suite**

```
bash
# Execute all tests
cd scripts/tests/notification-system
npx ts-node run-all-tests.ts
```
- Review Results**
 - Check generated test report

- Fix any failing tests
- Document edge cases

Short-term Improvements

1. Mock Email Service

- Create email mock for unit tests
- Use real SMTP only for integration tests
- Speed up test execution

2. Test Data Management

- Create test user setup script
- Add database seeding for tests
- Implement test cleanup

3. Performance Testing

- Add load tests for API endpoints
- Test concurrent notification processing
- Measure email delivery time

Long-term Strategy

1. Automated Testing

- Set up GitHub Actions
- Run tests on every commit
- Automated deployment on test pass

2. Monitoring

- Add test coverage reporting
- Track test execution time
- Alert on test failures

3. Documentation

- Keep test documentation updated
- Add inline comments for complex tests
- Create troubleshooting guide

Appendix

Test Helper Functions






Each test module includes:

- `runTest()` - Execute test with timing and error handling
- `getTestUserToken()` - Retrieve authenticated test user token
- Test result tracking and reporting

Test Result Format

```
interface TestResult {  
  name: string;    // Test description  
  passed: boolean; // Pass/fail status  
  error?: string;  // Error message if failed  
  duration: number; // Execution time in ms  
}
```

Color-Coded Output

-  Green: Passed tests
-  Red: Failed tests
-  Yellow: Warnings
-  Blue: Information
-  Magenta: Test suite headers

Support

For issues or questions about the test suite:

1. Check this documentation
2. Review test module comments
3. Consult main notification system documentation
4. Contact development team

Test Suite Version 1.0

Last Updated: December 3, 2025