Fleet Management Application - Complete Documentation

Application Overview

Scope: x_1554358_fleet

Application Name: Fleet Management

Purpose: Comprehensive university fleet vehicle reservation system with automated workflows, service catalog integration, and professional service portal

---

System Architecture

Core Tables

1. Fleet Operations (x_1554358_fleet_fleet_ops)

Purpose: Main table for vehicle reservation requests

Key Fields:

- `number` - Auto-generated request number
- `requested_for` - Reference to requesting user
- `vehicle_type` - Reference to Type table
- `make` - Reference to Make table
- `model` - Reference to Vehicle table (actual vehicle selection)
- `reservation_begin/end` - Planned reservation dates
- `u_date_returned` - Actual return date (auto-populated)
- `destination` - Trip destination
- `description` - Purpose of trip
- `reservation_status` - Choice: requested, checked_out, returned, closed_complete
- `assignment_group` - Auto-assigned to Fleet Management

Vehicle Information Section (Fleet Staff Only):

- `mileage` - Starting mileage (auto-populated from vehicle)
- `u_mileage_current` - Ending mileage (from inspection)
- `u_miles_driven` - Calculated miles driven

- `u_fuel` - Fuel added during trip (gallons)
- `u_mpg` - Calculated miles per gallon
- `u_visual_inspection_notes` - Notes from vehicle inspection

2. Vehicle (x_1554358_fleet_vehicle)

Purpose: Individual vehicle records

Key Fields:

- `vehicle_model` - Reference to Model table
- `vehicle_type` - Reference to Type table (auto-populated)
- `make` - Reference to Make table (auto-populated)
- `year` - Vehicle year (auto-populated)
- `mileage` - Current odometer reading
- `u_vehicle_status` - Choice: available, reserved, maintenance
- `available` - Boolean sync with status (for reference qualifier)
- `vehicle_image` - Image field for vehicle photos
- Vehicle details: exterior, interior, engine, VIN, etc.

3. Fleet Inspection Task (x_1554358_fleet_fleet_inspection_task)

Purpose: Extends Task table for vehicle return inspections

Key Fields:

- Standard task fields (state, assignment_group, etc.)
- `u_fleet_request` - Reference back to Fleet Operations
- `u_inspection_mileage` - Current odometer reading
- `u_fuel_added` - Fuel added (gallons)
- `u_visual_inspection_complete` - Mandatory checkbox
- `u_visual_inspection_notes` - Inspection notes

4. Reference Tables

- **Type** (x_1554358_fleet_type): Sedan, SUV, Van, Truck, Bus, Other
- **Make** (x_1554358_fleet_make): Manufacturer with type (e.g., "Toyota (SUV)")
- **Model** (x_1554358_fleet_model): Specific models with make/type/year references

Business Logic Implementation

Business Rules (12 Total)

1. **Auto-assign Fleet Requests**
   - Table: x_1554358_fleet_fleet_ops
   - When: after insert
   - Action: Assigns new requests to Fleet Management group
2. **Populate Vehicle Information**
   - Table: x_1554358_fleet_fleet_ops
   - When: before insert/update
   - Action: Auto-populates starting mileage from vehicle record
3. **Update Vehicle Status on Checkout**
   - Table: x_1554358_fleet_fleet_ops
   - When: after update
   - Action: Updates vehicle status based on reservation status
4. **Set Date Returned**
   - Table: x_1554358_fleet_fleet_ops
   - When: before insert/update
   - Action: Auto-populates return date when status = returned
5. **Create Return Task**
   - Table: x_1554358_fleet_fleet_ops
   - When: after update
   - Action: Creates inspection task when status = returned
6. **Complete Fleet Request and Release Vehicle**
   - Table: x_1554358_fleet_fleet_inspection_task
   - When: after update
   - Action: Syncs inspection data, calculates MPG, sends email, updates vehicle
7. **Validate Fleet Inspection Data**
   - Table: x_1554358_fleet_fleet_inspection_task
   - When: before update

    - Action: Validates mileage readings and inspection completion

8. **Sync Available Boolean Field**
    - Table: x_1554358_fleet_vehicle
    - When: before insert/update
    - Action: Keeps boolean field in sync with status choice field
9. **Validate Reservation Dates**
    - Table: x_1554358_fleet_fleet_ops
    - When: before insert/update
    - Action: Ensures end date > begin date, prevents past dates
10. **Check Vehicle Availability**
    - Table: x_1554358_fleet_fleet_ops
    - When: before insert/update
    - Action: Prevents booking unavailable vehicles
11. **SaveVehicleForm (Auto-fill Type, Make & Year)**
    - Table: x_1554358_fleet_vehicle
    - When: before insert/update
    - Action: Auto-populates from model selection
12. **Prevent Duplicate Vehicle Model**
    - Table: x_1554358_fleet_vehicle
    - When: before insert/update
    - Action: Ensures one vehicle per model

Client Scripts & UI Policies

Client Scripts:

- Auto-fill Type, Make & Year on Vehicle form
- Auto-populate Type from Make selection
- Hide Vehicle Information from regular users
- Auto-populate Date Returned
- Clear dependent fields

UI Policies:

- Lock fields after model selection

- Show Date Returned field conditionally
- Show 'Other' field when Type is 'Other'

Security & Access Control

Role-Based Access

Regular Users (user role):

- Create fleet requests
- View own requests only
- Vehicle Information section hidden
- Reservation status read-only after submission

Fleet Staff (Fleet Management, Fleet Motorpool, Fleet Admins):

- View/manage all fleet requests
- Change reservation status throughout lifecycle
- Complete vehicle inspections
- Access Vehicle Information section
- Manage vehicle, make, model, type records

Access Control Lists (ACLs)

Fleet Operations Table:

- Read: Fleet staff see all, users see own records only
- Write: Fleet staff modify all, users modify own if status=requested
- Create: Any user with "user" role
- Delete: Fleet admins only

Reference Tables:

- Read: All users (for dropdowns)
- Write/Create: Fleet management only
- Delete: Fleet admins only

Vehicle Table:

- All operations: Fleet staff only

Fleet Inspection Task Table:

- All operations: Fleet staff only

Service Catalog Integration

Record Producer Configuration

Name: Fleet Vehicle Reservation

Category: Transportation Services
Variables:

- `vehicle_type` - Select Box (Sedan, SUV, Van, Truck, Bus, Other)
- `reservation_begin` - Date/Time field
- `reservation_end` - Date/Time field
- `destination` - String field
- `description` - Multi Line Text (Purpose of trip)
- `short_description` - String (Special Requirements)

Automated Workflow

Process Flow:

1. User submits catalog request
2. Record Producer automatically creates Fleet Operations record
3. All existing business rules fire automatically
4. Request auto-assigned to Fleet Management group
5. Fleet staff processes using standard workflow

Benefits:

- No workflow creation needed
- Seamless integration with existing automation
- Self-service user experience
- Maintains all business logic

Service Portal Implementation

Portal Structure

Portal Name: Fleet Management Portal
URL: `https://[instance]/fleet`
Theme: Stock (with custom navigation)

Portal Pages

1. Calendar Page (Homepage)

URL: `/fleet?id=fleet_calendar`
Widget: Fleet Reservations Calendar
Features:

- Interactive calendar grid showing all active reservations
- Monthly navigation (Previous/Next)
- Color-coded reservation blocks
- Click for reservation details
- Upcoming reservations section (future reservations only)

2. My Reservations Page

URL: `/fleet?id=my_reservations`
Widget: My Fleet Reservations
Features:

- User's personal reservation history
- Status-based color coding
- Detailed reservation information
- Trip summary data (miles driven, MPG)

3. New Reservation Page

URL: `/fleet?id=new_reservation`
Widget: New Fleet Reservation
Features:

- Clean interface with trip requirements
- Direct link to Service Catalog item

- Quick access to reservation status

4. Vehicle Gallery Page

URL: `/fleet?id=vehicle_gallery`
Widget: Fleet Vehicle Gallery
Features:

- Grid layout with vehicle images
- Click for detailed vehicle information
- Status indicators (available, reserved, maintenance)
- Hover effects for interactivity

Navigation System

Widget: Fleet Navigation Bar
Implementation: Custom widget added to all pages
Features:

- Consistent navigation across all pages
- Active page highlighting
- Responsive design
- Simple URL-based page detection

Widget Architecture

1. Fleet Reservations Calendar Widget

Server Script:

- Queries Fleet Operations for non-closed reservations
- Separates all reservations (calendar) from upcoming (list)
- Uses string-based date comparison for reliability

Client Script:

- Builds calendar grid with 7-day weeks
- Handles month navigation
- Maps reservations to calendar days

- Provides click handlers for reservation details

HTML Template:

- Calendar header with month navigation
- 7-column grid layout for calendar days
- Reservation blocks with request numbers and user names
- Upcoming reservations list

2. My Fleet Reservations Widget

Server Script:

- Queries user-specific reservations
- Retrieves complete reservation history
- Includes trip data (miles driven, fuel, etc.)

HTML Template:

- Card-based layout for each reservation
- Status badges with color coding
- Detailed trip information
- Responsive grid design

3. Fleet Vehicle Gallery Widget

Server Script:

- Queries all vehicle records
- Includes vehicle images and specifications
- Status and availability information

Client Script:

- Handles vehicle detail popups
- Provides click interaction

HTML Template:

- Grid layout for vehicle cards
- Image display with fallback for missing photos

- Hover effects for better UX
- Vehicle information overlay

4. Fleet Navigation Bar Widget

Client Script:

- URL parsing to determine current page
- Sets active page for highlighting

HTML Template:

- Navigation links to all portal pages
- Conditional CSS classes for active states

Data Flow Examples

Complete Reservation Lifecycle

1. **User Submission:**
    - User submits Service Catalog request
    - Record Producer creates Fleet Operations record
    - Status: "requested", Vehicle: "available"
    - Auto-assigned to Fleet Management group
2. **Fleet Processing:**
    - Fleet staff changes status to "checked_out"
    - Vehicle status updates to "reserved"
    - Starting mileage populated from vehicle record
3. **Vehicle Return:**
    - Fleet staff changes status to "returned"
    - Vehicle status updates to "maintenance"
    - Inspection task created automatically
4. **Inspection Completion:**
    - Fleet staff completes inspection task
    - Mileage and fuel data synced to Fleet Operations
    - MPG calculated automatically

    - Email sent to user with trip summary
    - Vehicle status returns to "available"
    - Request status changed to "closed_complete"

Email Notification Variables

- `${requested_for.first_name}` - User's name
- `${number}` - Request number
- `${model}` - Vehicle model
- `${reservation_begin}` to `${reservation_end}` - Dates
- `${destination}` - Trip destination
- `${u_miles_driven}` - Calculated miles
- `${u_fuel}` - Fuel used
- `${u_mpg}` - Calculated efficiency

Technical Implementation Notes

Key Lessons Learned

Reference Qualifiers:

- Choice field filtering doesn't work reliably in reference qualifiers
- Multiple dependent fields cause issues; use single dependency
- Boolean field filtering works where choice field filtering fails

Extended Tables:

- Extending Task table requires field-level ACLs for inherited fields
- Extended tables provide clean separation from base ServiceNow tables

Role Management:

- Scoped application roles need proper naming (x_scope.role_name)
- Groups need appropriate base roles for functionality

Business Rule Patterns:

- Use `current.field.changes()` for field change detection

- Server-side validation more reliable than client-side
- Email notifications work better with `gs.eventQueue()`

Data Integrity:

- Always handle null values with `|| ''` or `|| 0`
- Server-side Business Rules needed for data persistence
- Client scripts provide immediate feedback, Business Rules ensure saving

Portal Development:

- Widget-based CSS often more maintainable than stylesheets
- String-based date comparisons more reliable than JavaScript Date parsing
- Simple navigation widgets work better than theme-dependent menus

---

Production Deployment

Testing Checklist

☐ Regular user can create requests
☐ Fleet staff can manage complete lifecycle
☐ Vehicle status updates correctly
☐ Email notifications send with correct data
☐ Reference qualifiers filter properly
☐ Role-based visibility works
☐ Data validation prevents invalid entries
☐ Service Portal pages load correctly
☐ Navigation highlighting functions
☐ Vehicle images display properly

Performance Considerations

- Business rules optimized for single-record operations
- Calendar widget limits date ranges for performance
- Vehicle gallery uses efficient image loading
- Portal pages use appropriate data limiting

Support Information

Key Tables for Troubleshooting:

- System Logs - Business Rule execution
- Email Logs - Notification delivery
- ACL debugging - Access issues
- Portal debugging - Widget and navigation issues

Common Issues:

- Tasks not visible: Check Fleet admin role assignment
- Reference qualifiers not working: Verify field configuration
- Email not sending: Check Event Registration and template
- Portal navigation not highlighting: Check client script URL parsing
- Vehicle images not displaying: Verify image field name (vehicle_image)

Future Enhancement Opportunities

Reporting Enhancements

- Vehicle utilization reports
- User activity summaries
- Fuel efficiency tracking
- Cost analysis dashboards
- Predictive maintenance scheduling

Feature Additions

- Mobile app for inspections
- Calendar system integration
- Fuel card API integration
- Advanced approval workflows
- Multi-location support

Technical Improvements

- Advanced caching for portal performance
- Automated data archiving
- Enhanced mobile responsiveness

- Integration with external fleet management systems

Conclusion

This Fleet Management Application represents a complete, production-ready solution that transforms vehicle reservation from a manual process into an automated, data-driven system. The integration of Service Catalog self-service capabilities with a professional Service Portal provides users with an intuitive interface while maintaining comprehensive backend automation for fleet staff.

The system successfully balances user experience simplicity with administrative power, providing a robust platform for efficient fleet management while maintaining data integrity and security throughout the entire vehicle lifecycle.