ODK MCP System Examples

This directory contains example workflows and use cases for the ODK MCP System.

Table of Contents

Basic Workflow

This example demonstrates the basic workflow of the ODK MCP System:

1. Create a project
2. Create a form
3. Collect data
4. Analyze data
5. Generate a report

Step 1: Create a Project

1. Sign in to the ODK MCP System
2. Navigate to the Projects section
3. Click "New Project"
4. Fill in the project details:
5. Name: "Basic Workflow Example"
6. Description: "A simple example of the ODK MCP System workflow"
7. Click "Create Project"

Step 2: Create a Form

1. Navigate to the Forms section
2. Click "New Form"
3. Click "Upload XLSForm"
4. Download the example form
5. Upload the form
6. Enter the form name: "Basic Form"
7. Click "Upload"

The example form includes the following fields: - Name (text) - Age (integer) - Gender (select_one: Male, Female, Other) - Education (select_one: Primary, Secondary, Higher) - Income (decimal) - Satisfaction (select_one: Very Satisfied, Satisfied, Neutral, Dissatisfied, Very Dissatisfied) - Comments (text)

Step 3: Collect Data

1. Navigate to the Data Collection section
2. Select the "Basic Form"
3. Click "Start New Submission"
4. Fill in the form with sample data
5. Click "Submit"
6. Repeat steps 3-5 to create multiple submissions

Alternatively, you can use the bulk submission feature:

1. Download the sample data
2. Navigate to the Data Collection section
3. Select the "Basic Form"
4. Click "Bulk Upload"
5. Upload the sample data file
6. Click "Submit"

Step 4: Analyze Data

1. Navigate to the Data Analysis section
2. Click on the "Descriptive Analytics" tab
3. Select the "Basic Form"
4. Select all variables
5. Click "Generate Analysis"
6. View the summary statistics and visualizations

For inferential statistics:

1. Click on the "Inferential Statistics" tab
2. Select the "Basic Form"
3. Choose "t-test" as the analysis type
4. Select "Age" as the variable
5. Select "Gender" as the group variable
6. Click "Run Analysis"
7. View the results

Step 5: Generate a Report

1. Navigate to the Reports section
2. Click "New Report"
3. Select "Standard Report"
4. Fill in the report details:
5. Title: "Basic Workflow Report"
6. Description: "A report of the basic workflow example"
7. Data Source: Select the analyses created in Step 4
8. Click "Generate Report"
9. View the report
10. Click "Export" to download the report as PDF

NGO Field Survey

This example demonstrates how an NGO might use the ODK MCP System for a field survey.

Scenario

An NGO is conducting a survey to assess the needs of a community after a natural disaster. They need to collect data on: - Household demographics - Damage assessment - Immediate needs - Long-term recovery plans

Implementation

1. Create a project named "Disaster Response Survey"
2. Create a form using the disaster response form template
3. Train field workers to use the system for data collection
4. Collect data in the field, using offline mode when necessary
5. Synchronize data when internet connectivity is available
6. Analyze the data to identify priority areas for intervention

7. Generate reports for donors and stakeholders

Analysis Examples

1. Descriptive statistics of damage levels by area
2. Correlation between household size and immediate needs
3. Prioritization of areas based on damage severity and vulnerability
4. Time-series analysis of recovery progress

Report Examples

1. Donor Report: Summary of findings and intervention priorities
2. Community Report: Simplified presentation of survey results for community feedback
3. Technical Report: Detailed analysis for program planning

Think Tank Research Project

This example demonstrates how a think tank might use the ODK MCP System for a research project.

Scenario

A think tank is conducting research on the impact of a new policy on small businesses. They need to collect data on: - Business demographics - Policy awareness - Implementation challenges - Economic impact - Future outlook

Implementation

1. Create a project named "Policy Impact Assessment"
2. Create a form using the policy impact form template
3. Collect data through interviews with business owners
4. Analyze the data to assess policy impact
5. Generate reports for policymakers and stakeholders

Analysis Examples

1. Comparative analysis of policy impact by business size
2. Regression analysis of factors affecting policy implementation
3. Cluster analysis to identify patterns in implementation challenges
4. Predictive modeling of future economic impact

Report Examples

1. Policy Brief: Concise summary of findings for policymakers
2. Research Report: Comprehensive analysis for academic audience
3. Executive Summary: Key findings for business associations

CSR Impact Assessment

This example demonstrates how a CSR department might use the ODK MCP System for impact assessment.

Scenario

A corporation is assessing the impact of its CSR initiatives in education. They need to collect data on: - School demographics - Program implementation - Student outcomes - Teacher feedback - Community perception

Implementation

1. Create a project named "Education CSR Impact Assessment"
2. Create a form using the CSR impact form template
3. Collect data from schools, teachers, students, and community members
4. Analyze the data to assess program impact
5. Generate reports for corporate leadership and stakeholders

Analysis Examples

1. Before-after comparison of student outcomes
2. Correlation between implementation fidelity and outcomes
3. Cost-benefit analysis of different program components
4. Qualitative analysis of teacher and community feedback

Report Examples

1. Executive Dashboard: Key metrics and visualizations for leadership
2. Impact Report: Comprehensive assessment for stakeholders
3. Program Improvement Plan: Recommendations based on findings

Health Monitoring Program

This example demonstrates how the ODK MCP System might be used for a health monitoring program.

Scenario

A health organization is monitoring the prevalence of a disease in a region. They need to collect data on: - Patient demographics - Symptoms and diagnosis - Treatment adherence - Outcomes - Environmental factors

Implementation

1. Create a project named "Disease Surveillance Program"
2. Create a form using the health monitoring form template
3. Train health workers to collect data during patient visits
4. Analyze the data to track disease prevalence and outcomes
5. Generate reports for health authorities and program managers

Analysis Examples

1. Geospatial analysis of disease prevalence
2. Time-series analysis of disease trends
3. Risk factor analysis using logistic regression
4. Treatment effectiveness analysis

Report Examples

1. Weekly Surveillance Report: Current disease status and trends
2. Quarterly Program Report: Comprehensive analysis of program performance
3. Annual Epidemiological Report: In-depth analysis of disease patterns and risk factors

Education Assessment

This example demonstrates how the ODK MCP System might be used for an education assessment.

Scenario

An education department is assessing the quality of schools in a district. They need to collect data on: - School infrastructure - Teacher qualifications and attendance - Student enrollment and attendance - Learning outcomes - Parent satisfaction

Implementation

1. Create a project named "School Quality Assessment"
2. Create a form using the education assessment form template
3. Train assessors to collect data during school visits
4. Analyze the data to assess school quality
5. Generate reports for education authorities and school administrators

Analysis Examples

1. Composite school quality index calculation
2. Correlation between infrastructure and learning outcomes
3. Comparative analysis of schools by location and type
4. Trend analysis of school performance over time

Report Examples

1. School Profile: Individual assessment results for each school
2. District Report: Comparative analysis of all schools in the district
3. Improvement Plan: Recommendations based on assessment findings

Environmental Monitoring

This example demonstrates how the ODK MCP System might be used for environmental monitoring.

Scenario

An environmental organization is monitoring water quality in a watershed. They need to collect data on: - Sampling location - Physical parameters (temperature, turbidity) - Chemical parameters (pH, dissolved oxygen) - Biological indicators - Land use in the surrounding area

Implementation

1. Create a project named "Watershed Monitoring Program"

2. Create a form using the environmental monitoring form template
3. Train volunteers to collect water samples and record data
4. Analyze the data to assess water quality
5. Generate reports for environmental authorities and community stakeholders

Analysis Examples

1. Water quality index calculation
2. Geospatial analysis of pollution hotspots
3. Correlation between land use and water quality
4. Trend analysis of water quality over time

Report Examples

1. Monthly Monitoring Report: Current water quality status
2. Quarterly Watershed Report: Comprehensive analysis of water quality trends
3. Annual State of the Watershed Report: In-depth analysis of watershed health

Custom Analysis Examples

This section provides examples of custom analyses that can be performed using the ODK MCP System.

Python Script Example

```
# Custom analysis script for income distribution
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# Load data
data = pd.read_csv('income_data.csv')

# Calculate income statistics
income_mean = data['income'].mean()
income_median = data['income'].median()
income_std = data['income'].std()
income_min = data['income'].min()
income_max = data['income'].max()

# Calculate income quintiles
quintiles = np.percentile(data['income'], [20, 40, 60, 80])
data['income_quintile'] = pd.qcut(data['income'], 5,
labels=['Q1', 'Q2', 'Q3', 'Q4', 'Q5'])
```

```python
# Calculate Gini coefficient
def gini(array):
    """Calculate the Gini coefficient of a numpy array."""
    # Mean absolute difference
    mad = np.abs(np.subtract.outer(array, array)).mean()
    # Relative mean absolute difference
    rmad = mad / np.mean(array)
    # Gini coefficient
    g = 0.5 * rmad
    return g

gini_coefficient = gini(data['income'].values)

# Generate visualizations
plt.figure(figsize=(12, 8))

# Income distribution
plt.subplot(2, 2, 1)
plt.hist(data['income'], bins=30, edgecolor='black')
plt.title('Income Distribution')
plt.xlabel('Income')
plt.ylabel('Frequency')

# Lorenz curve
plt.subplot(2, 2, 2)
lorenz = np.cumsum(np.sort(data['income'])) /
data['income'].sum()
plt.plot(np.linspace(0, 1, len(lorenz)), lorenz)
plt.plot([0, 1], [0, 1], 'r--')
plt.title(f'Lorenz Curve (Gini = {gini_coefficient:.3f})')
plt.xlabel('Cumulative share of population')
plt.ylabel('Cumulative share of income')

# Income by gender
plt.subplot(2, 2, 3)
gender_income = data.groupby('gender')
['income'].mean().reset_index()
plt.bar(gender_income['gender'], gender_income['income'])
plt.title('Average Income by Gender')
plt.xlabel('Gender')
plt.ylabel('Average Income')

# Income by education
plt.subplot(2, 2, 4)
edu_income = data.groupby('education')
['income'].mean().reset_index()
plt.bar(edu_income['education'], edu_income['income'])
plt.title('Average Income by Education')
plt.xlabel('Education Level')
plt.ylabel('Average Income')
plt.xticks(rotation=45)
```

```python
plt.tight_layout()
plt.savefig('income_analysis.png')

# Output results
results = {
    'income_statistics': {
        'mean': income_mean,
        'median': income_median,
        'std': income_std,
        'min': income_min,
        'max': income_max,
        'quintiles': quintiles.tolist(),
        'gini_coefficient': gini_coefficient
    },
    'visualizations': ['income_analysis.png']
}

print(results)
```

R Script Example

```r
 # Custom analysis script for survey data
library(tidyverse)
library(survey)
library(ggplot2)

# Load data
data <- read.csv("survey_data.csv")

# Create survey design object
survey_design <- svydesign(
  ids = ~cluster_id,
  strata = ~stratum,
  weights = ~weight,
  data = data
)

# Calculate weighted means
weighted_means <- svymean(~age + income + education,
survey_design)

# Calculate weighted proportions
weighted_props <- svymean(~factor(gender) +
factor(satisfaction), survey_design)

# Logistic regression model
model <- svyglm(
  satisfaction_binary ~ age + income + factor(education) +
factor(gender),
```

```r
  family = quasibinomial(),
  design = survey_design
)

# Generate visualizations
# Weighted satisfaction by education
satisfaction_by_edu <- svyby(
  ~satisfaction_score,
  ~education,
  survey_design,
  svymean
)

pdf("survey_analysis.pdf")

ggplot(satisfaction_by_edu, aes(x = education, y =
satisfaction_score)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = satisfaction_score - 1.96 * se, ymax
= satisfaction_score + 1.96 * se), width = 0.2) +
  labs(title = "Satisfaction Score by Education Level",
       x = "Education Level",
       y = "Average Satisfaction Score") +
  theme_minimal()

dev.off()

# Output results
results <- list(
  weighted_means = weighted_means,
  weighted_props = weighted_props,
  model_summary = summary(model),
  visualizations = "survey_analysis.pdf"
)

print(results)
```

SQL Query Example

```sql
 -- Custom SQL query for data exploration
WITH submission_counts AS (
  SELECT
    DATE(submitted_at) AS submission_date,
    COUNT(*) AS submission_count
  FROM
    submissions
  WHERE
    form_id = 123
    AND project_id = 456
    AND submitted_at >= DATE('now', '-30 days')
```

```
  GROUP BY
    DATE(submitted_at)
),
daily_stats AS (
  SELECT
    DATE(submitted_at) AS submission_date,
    AVG(CAST(JSON_EXTRACT(data, '$.age') AS INTEGER)) AS
avg_age,
    COUNT(DISTINCT submitted_by) AS unique_submitters
  FROM
    submissions
  WHERE
    form_id = 123
    AND project_id = 456
    AND submitted_at >= DATE('now', '-30 days')
  GROUP BY
    DATE(submitted_at)
)
SELECT
  sc.submission_date,
  sc.submission_count,
  ds.avg_age,
  ds.unique_submitters,
  CASE
    WHEN LAG(sc.submission_count) OVER (ORDER BY
sc.submission_date) IS NULL THEN 0
    ELSE (sc.submission_count - LAG(sc.submission_count) OVER
(ORDER BY sc.submission_date)) * 100.0 /
LAG(sc.submission_count) OVER (ORDER BY sc.submission_date)
  END AS submission_growth_pct
FROM
  submission_counts sc
JOIN
  daily_stats ds ON sc.submission_date = ds.submission_date
ORDER BY
  sc.submission_date;
```

Integration Examples

This section provides examples of integrating the ODK MCP System with other tools and services.

Baserow Integration Example

```
import requests
import json

# ODK MCP System API
```

```
odk_base_url = "http://localhost:8000/api/v1"
odk_token = "your_odk_api_token"

# Baserow API
baserow_url = "https://baserow.example.com/api"
baserow_token = "your_baserow_api_token"

# Configure Baserow integration
response = requests.post(
    f"{odk_base_url}/data-aggregation/integrations/baserow/
configure",
    json={
        "url": baserow_url,
        "api_token": baserow_token,
        "enabled": True
    },
    headers={
        "Authorization": f"Bearer {odk_token}"
    }
)
print(response.json())

# Create a table in Baserow for survey data
response = requests.post(
    f"{baserow_url}/database/tables/",
    json={
        "database_id": 1,
        "name": "Survey Data"
    },
    headers={
        "Authorization": f"Token {baserow_token}"
    }
)
table_id = response.json()["id"]
print(f"Created table with ID: {table_id}")

# Create fields in the Baserow table
fields = [
    {"name": "Name", "type": "text"},
    {"name": "Age", "type": "number"},
    {"name": "Gender", "type": "single_select",
"select_options": ["Male", "Female", "Other"]},
    {"name": "Education", "type": "single_select",
"select_options": ["Primary", "Secondary", "Higher"]},
    {"name": "Income", "type": "number",
"number_decimal_places": 2},
    {"name": "Satisfaction", "type": "single_select",
"select_options": ["Very Satisfied", "Satisfied", "Neutral",
"Dissatisfied", "Very Dissatisfied"]},
    {"name": "Comments", "type": "long_text"}
]
```

```
for field in fields:
    field_type = field.pop("type")
    select_options = field.pop("select_options", None)

    response = requests.post(
        f"{baserow_url}/database/fields/table/{table_id}/",
        json={
            "name": field["name"],
            "type": field_type,
            **field
        },
        headers={
            "Authorization": f"Token {baserow_token}"
        }
    )
    field_id = response.json()["id"]
    print(f"Created field {field['name']} with ID: {field_id}")

    if select_options:
        for option in select_options:
            response = requests.post(
                f"{baserow_url}/database/fields/{field_id}/
select-options/",
                json={
                    "value": option,
                    "color": "blue"
                },
                headers={
                    "Authorization": f"Token {baserow_token}"
                }
            )
            print(f"Added option {option} to field
{field['name']}")

# Sync data from ODK MCP System to Baserow
response = requests.post(
    f"{odk_base_url}/data-aggregation/integrations/baserow/
sync",
    json={
        "project_id": 456,
        "form_id": 123,
        "table_id": table_id,
        "field_mapping": {
            "name": "Name",
            "age": "Age",
            "gender": "Gender",
            "education": "Education",
            "income": "Income",
            "satisfaction": "Satisfaction",
            "comments": "Comments"
        }
    },
```

```
    headers={
        "Authorization": f"Bearer {odk_token}"
    }
)
print(response.json())
```

AI Tool Integration Example

```
import requests
import json

# ODK MCP System API
odk_base_url = "http://localhost:8000/api/v1"
odk_token = "your_odk_api_token"

# Configure Claude integration
response = requests.post(
    f"{odk_base_url}/data-aggregation/integrations/ai-tool/
configure",
    json={
        "tool": "claude",
        "api_key": "your_claude_api_key",
        "model": "claude-3-opus-20240229",
        "enabled": True
    },
    headers={
        "Authorization": f"Bearer {odk_token}"
    }
)
print(response.json())

# Generate AI analysis
response = requests.post(
    f"{odk_base_url}/data-aggregation/integrations/ai-tool/
analyze",
    json={
        "project_id": 456,
        "form_id": 123,
        "prompt": """
        Analyze the survey data and provide insights on the
following:
        1. Key demographic patterns
        2. Factors affecting satisfaction levels
        3. Recommendations for improving satisfaction
        4. Any unexpected patterns or outliers in the data

        Please include visualizations where appropriate.
        """,
        "data_filter": {
            "submitted_at": {
```

```
                "operator": "gte",
                "value": "2023-01-01"
            }
        }
    },
    headers={
        "Authorization": f"Bearer {odk_token}"
    }
)
analysis_id = response.json()["data"]["analysis_id"]
print(f"Generated AI analysis with ID: {analysis_id}")

# Check analysis status
response = requests.get(
    f"{odk_base_url}/data-aggregation/integrations/ai-tool/
analyze/{analysis_id}",
    headers={
        "Authorization": f"Bearer {odk_token}"
    }
)
print(response.json())

# Include AI analysis in a report
response = requests.post(
    f"{odk_base_url}/data-aggregation/reports",
    json={
        "project_id": 456,
        "title": "AI-Enhanced Survey Analysis",
        "description": "Survey analysis with AI-generated
insights",
        "sections": [
            {
                "title": "AI Analysis",
                "ai_analysis_id": analysis_id
            }
        ],
        "format": "pdf",
        "options": {
            "include_cover_page": True,
            "include_table_of_contents": True
        }
    },
    headers={
        "Authorization": f"Bearer {odk_token}"
    }
)
report_id = response.json()["data"]["report_id"]
print(f"Created report with ID: {report_id}")
```

Advanced Features

This section demonstrates advanced features of the ODK MCP System.

Longitudinal Data Collection

This example demonstrates how to set up a longitudinal data collection project:

1. Create a project named "Longitudinal Study"
2. Create a baseline form using the baseline form template
3. Create a follow-up form using the follow-up form template
4. Link the forms using the "entity_id" field
5. Collect baseline data
6. Collect follow-up data at regular intervals
7. Analyze changes over time

Offline Data Collection

This example demonstrates how to use the offline data collection feature:

1. Enable offline mode in the Data Collection section
2. Download forms for offline use
3. Collect data without internet connectivity
4. Synchronize data when internet connectivity is restored

Custom Dashboards

This example demonstrates how to create custom dashboards:

1. Navigate to the Reports section
2. Click "New Report"
3. Select "Dashboard"
4. Add widgets to the dashboard:
5. Summary statistics
6. Charts and visualizations
7. Data tables
8. Maps
9. Configure refresh intervals for real-time data
10. Share the dashboard with stakeholders

Data Security Features

This example demonstrates the data security features of the ODK MCP System:

1. Configure user roles and permissions
2. Set up two-factor authentication
3. Configure data encryption
4. Set up audit logging
5. Configure data retention policies
6. Set up secure data sharing

API Integration

This example demonstrates how to integrate the ODK MCP System with other systems using the API:

1. Generate an API key in the Settings section
2. Use the API to:
3. Create forms
4. Submit data
5. Query data
6. Generate reports
7. Set up webhooks for real-time notifications
8. Implement custom integrations with other systems