

# Intelligent User Feedback Analysis and Action System

## Scenario

You are a product engineer at a B2C mobile application company managing a productivity app with approximately 10,000 active users. Your team currently receives daily feedback through multiple channels:

- 1 10–20 app store reviews per day
- 2 5–10 customer support emails per day
- 3 Occasional in-app feedback submissions

Currently, a team member manually reviews this feedback and creates tickets in the project management system. This manual workflow consumes 1–2 hours daily and often leads to:

- 1 Delayed responses to critical bugs
- 2 Inconsistent ticket formatting and prioritization
- 3 Lost or overlooked user feedback
- 4 Poor traceability from user complaint to engineering resolution

## Your Task: Complete Project Implementation

Design, implement, and demonstrate a complete multi-agent AI system that:

- 1 Reads user feedback from CSV files containing app store reviews and support emails
- 2 Classifies feedback into categories (Bug, Feature Request, Praise, Complaint, Spam)
- 3 Extracts actionable insights and relevant technical details
- 4 Generates structured tickets and logs them to CSV files with appropriate priorities and metadata
- 5 Ensures quality, consistency, and accuracy through automated review
- 6 Provides a user interface for monitoring, configuration, and manual overrides

## System Objectives

- 1 Automation: End-to-end processing without manual intervention
- 2 Speed: Feedback analysis and ticket creation within minutes
- 3 Consistency: Standardized ticket structure and priority assignment
- 4 Traceability: Clear linkage from original feedback to generated tickets
- 5 Usability: An intuitive dashboard for monitoring and control

## Multi-Agent Architecture

The system is built using a modular multi-agent architecture. Each agent has a clearly defined responsibility such as CSV ingestion, feedback classification, bug analysis, feature extraction, ticket creation, and quality validation.

- 1 CSV Reader Agent: Reads and parses feedback data from CSV files
- 2 Feedback Classifier Agent: Categorizes feedback using NLP
- 3 Bug Analysis Agent: Extracts steps to reproduce, platform details, and severity
- 4 Feature Extractor Agent: Identifies feature requests and estimates user impact
- 5 Ticket Creator Agent: Generates structured tickets and logs them to CSV files
- 6 Quality Critic Agent: Reviews tickets for completeness and accuracy

## Technical Implementation Requirements

- 1 Agent Orchestration: CrewAI or AutoGen
- 2 User Interface: Streamlit for monitoring and manual overrides
- 3 Input: CSV files containing mock feedback data
- 4 Output: CSV files for generated tickets and logs
- 5 Error Handling: Robust exception handling and structured logging
- 6 Configuration: Adjustable classification thresholds and priority rules

## Input Data Specifications

1. app\_store\_reviews.csv

Columns: review\_id, platform, rating, review\_text, user\_name, date, app\_version

2. support\_emails.csv

Columns: email\_id, subject, body, sender\_email, timestamp, priority

3. expected\_classifications.csv

Columns: source\_id, source\_type, category, priority, technical\_details, suggested\_title

## System Implementation

- 1 Agent Classes with single, well-defined responsibilities
- 2 CSV ingestion, processing, and structured output generation
- 3 NLP-based classification with confidence scoring
- 4 Actionable and standardized ticket creation
- 5 Automated quality validation before final output

## User Interface

- 1 Dashboard for processed feedback and generated tickets
- 2 Configuration panel for priorities and thresholds
- 3 Manual override and approval workflow
- 4 Analytics with performance and accuracy metrics

## Output Files

- 1 generated\_tickets.csv – Final structured ticket output
- 2 processing\_log.csv – Detailed processing decisions and logs

- 3 metrics.csv – Accuracy, throughput, and performance metrics

## Demonstration and Testing

- 1 CSV data ingestion and preprocessing
- 2 Live multi-agent interaction and processing flow
- 3 Classification accuracy comparison with expected results
- 4 Ticket generation with complete metadata
- 5 User interface monitoring and control
- 6 Robust error handling and edge case validation