**Product Requirements Document (PRD)**

### **1. Overview**

#### **Product Name**: Correspondence Processing and Analysis System (CPAS)

#### **Product Vision**: To create a system that processes and analyzes letters and correspondence sent to local councils via existing channels (e.g., emails, paper letters), enabling efficient issue identification, tracking, and resolution without requiring user adoption of a new platform.

#### **Target Audience**: Local councils, community organizations, and policymakers.

### **2. Objectives**

* Automate the processing of letters and correspondence received via emails, paper, and other existing channels.
* Categorize and prioritize issues based on the content of the correspondence.
* Provide actionable insights to local councils for issue resolution.
* Improve transparency and accountability in addressing community concerns.
* Reduce manual effort in categorizing and prioritizing issues.

### **3. Key Features**

#### **3.1. Correspondence Ingestion**

* **Description**: Integrate with existing correspondence channels (e.g., email inboxes, postal mail scanning systems) to collect letters and correspondence.
* **Requirements**:
  + Accept letters in multiple formats (e.g., PDF, scanned images, plain text, emails).
  + Automatically extract text from scanned documents using OCR (Optical Character Recognition).
  + Store all correspondence in a centralized database.

#### **3.2. Automated Issue Categorization**

* **Description**: Use Natural Language Processing (NLP) to analyze letters and categorize them by issue type (e.g., Air Pollution, Traffic Congestion).
* **Requirements**:
  + Predefined list of issue categories (e.g., Environmental, Infrastructure, Social).
  + Machine learning model to improve categorization accuracy over time.
  + Manual override option for councils to reclassify letters if needed.

#### **3.3. Sentiment Analysis**

* **Description**: Analyze the sentiment of each letter (e.g., Negative, Neutral, Positive) to gauge the urgency and emotional tone.
* **Requirements**:
  + Sentiment scoring for each letter.
  + Highlight letters with highly negative sentiment for priority attention.

#### **3.4. Issue Tracking and Resolution**

* **Description**: Enable councils to track the status of issues identified from correspondence (e.g., Reported, In Progress, Resolved).
* **Requirements**:
  + Dashboard for councils to view and manage issues.
  + Real-time updates on issue status.
  + Notifications for new high-priority issues.

#### **3.5. Data Visualization**

* **Description**: Provide visual representations of issue data derived from correspondence.
* **Requirements**:
  + Interactive map showing issue hotspots.
  + Charts for issue frequency, severity, and resolution rates.
  + Filters for issue categories, locations, and timeframes.

#### **3.6. Reporting and Analytics**

* **Description**: Generate reports and insights for councils and stakeholders.
* **Requirements**:
  + Exportable reports in CSV, PDF, and JSON formats.
  + Trends and patterns analysis (e.g., most common issues, resolution times).
  + Customizable dashboards for different user roles (e.g., council members, administrators).

### **4. Functional Requirements**

#### **4.1. User Roles**

* **Residents**: Send letters and correspondence via existing channels (e.g., email, postal mail).
* **Council Members**: View, categorize, and track issues; generate reports.
* **Administrators**: Manage system settings, user roles, and data.

#### **4.2. Data Management**

* **Correspondence Data**: Store letters, extracted text, issue categories, and sentiment scores.
* **Issue Data**: Store details such as issue name, category, severity, frequency, location, and status.
* **User Data**: Store user profiles for council members and administrators.

#### **4.3. Security and Privacy**

* **Data Encryption**: Ensure all data is encrypted during transmission and storage.
* **User Authentication**: Implement secure login and password recovery mechanisms.
* **Access Control**: Restrict access to sensitive data based on user roles.

### **5. Non-Functional Requirements**

#### **5.1. Performance**

* **Response Time**: Ensure the platform processes letters and updates dashboards within 5 seconds.
* **Scalability**: Support up to 10,000 letters per day.

#### **5.2. Usability**

* **User Interface**: Intuitive and easy to navigate for council members and administrators.
* **Accessibility**: Comply with WCAG 2.1 standards for accessibility.

#### **5.3. Reliability**

* **Uptime**: Ensure 99.9% uptime.
* **Backup**: Regular data backups to prevent data loss.

### **6. Success Metrics**

* **Issue Identification Accuracy**: Percentage of letters correctly categorized by the system.
* **Issue Resolution Rate**: Percentage of issues resolved within a specified timeframe.
* **User Satisfaction**: Average rating from council members and administrators.
* **Processing Time**: Average time taken to process and categorize a letter.

### **7. Roadmap**

#### **Phase 1: MVP (3-6 Months)**

* Develop core features: Correspondence Ingestion, Automated Issue Categorization, and Sentiment Analysis.
* Launch web platform with basic analytics.

#### **Phase 2: Enhancements (6-12 Months)**

* Add Issue Tracking and Resolution features.
* Improve NLP models for better categorization and sentiment analysis.

#### **Phase 3: Advanced Analytics (12-18 Months)**

* Implement advanced analytics and reporting tools.
* Integrate AI for predictive analysis (e.g., identifying potential issue hotspots).

### **8. Risks and Mitigation**

#### **8.1. Data Privacy Concerns**

* **Risk**: Sensitive information in letters may be exposed.
* **Mitigation**: Implement strong data encryption and access controls.

#### **8.2. Low Accuracy in Categorization**

* **Risk**: The system may misclassify letters, leading to incorrect issue tracking.
* **Mitigation**: Continuously train the NLP model with labeled data and allow manual corrections.

#### **8.3. Technical Challenges**

* **Risk**: Delays in development due to technical complexities.
* **Mitigation**: Use agile development methodologies and conduct regular testing.

### **9. Stakeholders**

* **Residents**: Send letters and correspondence via existing channels.
* **Local Councils**: Use the platform to process, categorize, and resolve issues.
* **Community Organizations**: Collaborate on issue resolution and awareness campaigns.
* **Developers**: Build and maintain the platform.
* **Project Managers**: Oversee development and ensure timely delivery.

### **10. Appendix**

#### **10.1. Sample Issue Categories**

* Environmental: Air Pollution, Water Scarcity, Waste Disposal.
* Infrastructure: Potholes, Road Surface Damage, Aging Infrastructure.
* Social: Homelessness, Youth Unemployment, Mental Health Services.
* Economic: High Cost of Living, Decline in Local Businesses, Rising Energy Costs.

#### **10.2. Sample User Stories**

* As a council member, I want to view all letters categorized by issue type so that I can prioritize them.
* As an administrator, I want to generate a report on issue resolution rates so that I can assess performance.
* As a resident, I want to send a letter about a pothole via email so that the council can address it.

This PRD outlines a system that **works with existing correspondence channels** (e.g., emails, paper letters) without requiring users to adopt a new system. Let me know if you need further refinements or additional details!