**Product Description Document**

**1. Introduction**

**Purpose**

This project aims to develop an \*\*AI-powered complaint management system\*\* to automate the processing of customer complaints for online delivery services. By leveraging advanced \*\*Natural Language Processing (NLP)\*\* and decision-making algorithms, the system will extract key details, apply business-specific compensation rules, and resolve issues efficiently with minimal human intervention.

**Scope**

The system will automate complaints submitted via \*\*email\*\* and \*\*WhatsApp\*\*, streamlining information extraction, rule-based decision-making, and resolution processes. Manual complaint handling and chatbot-based interactions are explicitly out of scope.

**Target Audience**

Delivery service providers (e.g., Wolt, 10bis) aiming to:

- Reduce complaint resolution times.

- Enhance customer satisfaction through streamlined automation.

**2. Business Goals**

**Objective**

Deliver a scalable, user-friendly platform that optimizes complaint resolution workflows while improving customer satisfaction and reducing operational inefficiencies.

**Key Performance Indicators (KPIs):**

- Reduce average resolution time by **40%**.

- Achieve a 90% first-time resolution rate.

- Increase customer satisfaction ratings by 20% within six months of deployment.

**3. User Personas and Stories**

**Customer Service Representative (CSR)**

-Goals: Quickly resolve complaints with minimal manual input.

- Pain Points: High workload due to repetitive, time-consuming tasks.

- Expectations: Automated resolution suggestions and clear next steps.

- User Story: "As a CSR, I need automated suggestions for resolving complaints so I can process cases faster and focus on high-priority tasks."

**4. Functional Requirements**

Must-Have Features

* Complaint intake via email and WhatsApp.
* NLP-based extraction of complaint details (e.g., order ID, issue type).
* Application of predefined compensation rules.
* Automated resolution generation or follow-up requests for missing details.

Optional Features

* Sentiment analysis to prioritize critical complaints.
* Multi-channel support for platforms like SMS and web forms.

**5. Non-Functional Requirements**

* Performance: Process 90% of complaints within 2 seconds.
* Scalability: Handle up to 10,000 complaints daily.
* Security: Ensure GDPR-compliant data handling and storage.

**6. User Workflow**

1. Complaint Submission: Customers send complaints via email or WhatsApp.
2. Information Extraction: The NLP engine extracts critical details like order ID, complaint type, and sentiment.
3. Decision-Making: Predefined business rules determine the resolution (e.g., refund, escalation).
4. Resolution Communication: Customers receive a resolution or a follow-up request for additional details.

**7. Narrative**

"Imagine Sarah, a customer service representative at a leading delivery service, starting her day. Instead of wading through hundreds of emails, she opens her dashboard where the system has already sorted complaints, prioritized critical cases, and generated resolution suggestions. Sarah reviews, approves, and moves on to high-impact work. Meanwhile, her manager monitors real-time analytics to identify recurring delivery issues, ensuring proactive resolutions before complaints spike. Customers receive faster responses, boosting satisfaction and loyalty."

**8. UI/UX Design**

Wireframe Concept  
A three-column layout with:

1. Complaint overview.
2. Suggested resolutions and actions.
3. Escalation or override options.

Design Principles

* Intuitive interface with clear navigation.
* Mobile and desktop responsiveness.
* Accessibility features for visually impaired users.

9. Technical Specifications

* System Architecture: Modular design with components for complaint intake, NLP processing, and business rule execution.
* Integration: RESTful APIs for CRM systems and compensation rule repositories.
* Platform: Web-based with mobile support.

**10. Testing and QA Plan**

Testing Phases:

* Unit Testing: Ensure NLP accuracy (e.g., 95% extraction accuracy on key details).
* Integration Testing: Validate email and WhatsApp complaint intake.
* User Acceptance Testing (UAT): Conduct with pilot companies to gather feedback.

QA Measures:

* Automated regression tests post-deployment.
* Regular updates to improve NLP model accuracy and compliance.

**11. Project Timeline and Milestones**

Phase 1: Requirements Gathering (2 weeks)

* + Finalize KPIs and stakeholder interviews.

Phase 2: System Design and Prototyping (3 weeks)

* + Deliver wireframes and technical architecture.

Phase 3: Development (8 weeks)

* + Build and integrate core modules.

Phase 4: Testing and Deployment (3 weeks)

* + Conduct QA testing and deploy pilot.

**12. Risks and Assumptions**

Risks:

* Incorrect data extraction may delay resolutions.
* Resistance to change as users may prefer manual workflows.

Mitigation Strategies:

* Regular training of NLP models with diverse datasets.
* Stakeholder workshops to demonstrate system value.

**13. Appendices**

1. Literature Review and Competitor Analysis: Highlights superior speed and scalability compared to existing solutions like Zendesk AI or Freshworks Freddy AI.
2. Project Repo - <https://github.com/ShovalBenjer/Customer_Service_App>
3. Project Roadmap - https://github.com/users/ShovalBenjer/projects/4