LO1 Requirement Analysis

1. Stakeholder Analysis:

1.1 Stakeholders Identification:
Customers: Individuals ordering pizza via the PizzaDronz app.
Drone Operators: Individuals or teams responsible for managing drone operations.
Restaurants: Pizza providers partnering with PizzaDronz for delivery.
Project Team Members: Including developers, designers, project managers, and testers involved in the PizzaDronz project.
Project Sponsors: Investors or senior management funding the project.
Regulatory Bodies: Authorities regulating drone flights and food delivery services.
Community: People living in areas where PizzaDronz operates, who might be affected by drone flights.

1.2 Methods to Gather Requirements:
Interviews: One-on-one discussions with stakeholders to explore their needs and expectations.
Surveys and Questionnaires: Broad tools for collecting requirements from a larger audience.
Focus Groups: Bringing together a representative group from each stakeholder category to discuss their requirements.
Observation: Understanding stakeholder interactions with current systems or processes.
Document Analysis: Reviewing existing documentation of business processes and systems.

2. Requirements Collection:

2.1 Potential Requirements from Stakeholders:

Customers:
Fast and accurate pizza delivery.
Easy-to-use app interface for ordering.
Real-time tracking of pizza delivery.

Drone Operators:
Clear, real-time weather and no-fly zone information.
Safe and efficient drone navigation systems.

Restaurants:
Seamless integration with the PizzaDronz ordering system.
Timely updates on order status and delivery timings.

Project Team Members:
Clear project objectives and milestones.
Access to the latest development tools and technologies.

Project Sponsors
Return on investment (ROI) and project success metrics.
Regular progress reports and reviews.

Regulatory Bodies:
Compliance with drone flight regulations.
Ensuring food safety standards are maintained.

Community:
Minimizing noise pollution from drones.
Ensuring drones do not invade privacy.

By identifying these stakeholders and their potential requirements, the PizzaDronz project can develop a comprehensive understanding of the needs across its ecosystem. This will guide the project team in prioritizing features and functionalities that align with the expectations of all involved parties, ensuring a successful and well-received delivery system.

2.2 Use Case Analysis

Use Case One: Ordering Process
Scenario Description:
The customer browses through various restaurants and pizza menus on the PizzaDronz app on their smartphone.
They select their favorite restaurant and pick a pizza from the menu.
On the confirmation page, the customer checks the order details, selects the delivery address (Appleton Tower), and confirms the total order amount.
The customer clicks "Place Order" to proceed to the payment process.

Use Case Two: Payment
Scenario Description:
At the payment page, the customer chooses their payment method (credit card).
The customer enters their credit card information and confirms the security of the payment details.
The customer clicks "Pay", and the system processes the transaction.
Upon successful payment, the customer receives a notification confirming the order and payment success.

Use Case Three: Drone Delivery
Scenario Description:
Once the order and payment information are confirmed, the drone receives a delivery task.
The drone navigates to the customer's delivery address using preset flight paths and real-time obstacle avoidance technology.
After the customer confirms the receipt of the pizza, the drone returns to the Appleton Tower.

3. Requirements Analysis and Validation:

3.1 Requirements Analysis

Clarity:

All collected requirements must be articulate, specific, and unambiguous. This involves refining the descriptions of functional, performance, robustness, security, and usability requirements to eliminate any vagueness. For instance, the requirement for the system to "accurately process orders" should specify what constitutes accuracy in order processing, including order intake, confirmation, and delivery details.

Completeness:

The requirements must cover all aspects of the project, both functional and non-functional. This includes ensuring that all user interactions, system operations, and external integrations are fully described. For example, the performance requirements should not only mention "system response time" but also define acceptable response times for different operations.

Consistency:

Checking for contradictions among requirements to ensure they can coexist without conflict. For instance, the system's capability to "prevent unauthorized access and data breaches" must not contradict any usability requirements that advocate for ease of access to order information.

Feasibility:

Evaluating the practicality of implementing the requirements, considering technical, timeline, and cost aspects. For example, delivering pizzas within the set time constraints must be assessed against the current capabilities of drone technology and operational logistics.

3.2 Testing Strategy Assessment:

Unit Level Testing (White Box Testing):

Validates the internal logic and functionality of individual modules. For example, the drone module's pathfinding and obstacle avoidance features can be tested for algorithmic accuracy and response to simulated obstacles.

Integration Level Testing:

Ensures effective communication and cooperation between different system modules. The integration between the order system and the restaurant system can be tested to ensure that order information is accurately transmitted and received.

System Level Testing (Black Box Testing):

Assesses the complete PizzaDronz system's overall functionality and user experience. This includes testing the end-to-end process from order placement through to delivery and user feedback.

Statistical Tests:

Used to verify non-functional requirements like system performance metrics, including response

times. This helps to ensure that the system meets predefined performance standards.

This structure provides a clear delineation of requirements across different levels of the PizzaDronz system, from the overall system functionality to the specific operations of individual modules.

3.3 Potential Limitations:

One notable shortcoming in our testing approach is the lack of an effective method for testing security aspects, specifically the protection of user privacy information. While we can ensure system functionality and performance, ensuring data privacy and protection against breaches remains a challenge. This gap highlights the need for additional focus on security testing methodologies. This assessment demonstrates a broad coverage of testing across the system's various aspects but also acknowledges the necessity of enhancing the security testing domain.

This detailed approach to requirements analysis and validation, coupled with a comprehensive testing strategy, aims to cover the diverse aspects of the PizzaDronz project, ensuring a high-quality, secure, and user-friendly delivery system. However, the highlighted gap in security testing requires further attention to enhance the project's overall security posture.