

# EXCEPTION HANDLING

Date	06 NOVEMBER 2023
Team ID	NM2023TMID02250
Project Name	Project- Farmer Insurance Chain
Maximum Marks	4 Marks

Exception handling is crucial in a farmer insurance chain to ensure that the application can gracefully handle unexpected errors and failures. Here's a guide on how to approach exception handling in your software:

## 1. Identify Potential Exceptions:

Begin by identifying potential exceptions that can occur in your farmer insurance chain. This includes errors related to user input, external dependencies (e.g., databases or APIs), and internal logic.

## 2. Use Appropriate Exception Types:

Choose the appropriate exception types for different scenarios. In most programming languages, there are built-in exception classes and custom exceptions you can create to categorize errors more effectively.

### **3. Centralized Error Handling:**

Implement centralized error handling mechanisms, such as global error handlers or middleware, to capture and manage exceptions consistently throughout your application.

### **4. Throw Exceptions:**

Whenever an exceptional condition is encountered, throw an exception with a meaningful error message. Include relevant information about the error, such as the context and the cause.

### **5. Catch Exceptions:**

Use try-catch blocks or similar constructs to catch exceptions where they might occur. This allows you to handle the error gracefully without crashing the application.

### **6. Log Exceptions:**

Logging exceptions is crucial for debugging and troubleshooting. Log the exception details, including the type of exception, error message, and a stack trace, so you can trace the problem's origin.

### **7. Handle Gracefully:**

When catching an exception, decide how to handle it. This may involve displaying a user-friendly error message, attempting to recover from the error, or gracefully exiting the application when necessary.