**PRACTICAL - 3**

**AIM:** Create a model, define system architecture, and identify and mitigate potential threats using Microsoft Threat Modeling Tool.

**Solution:**

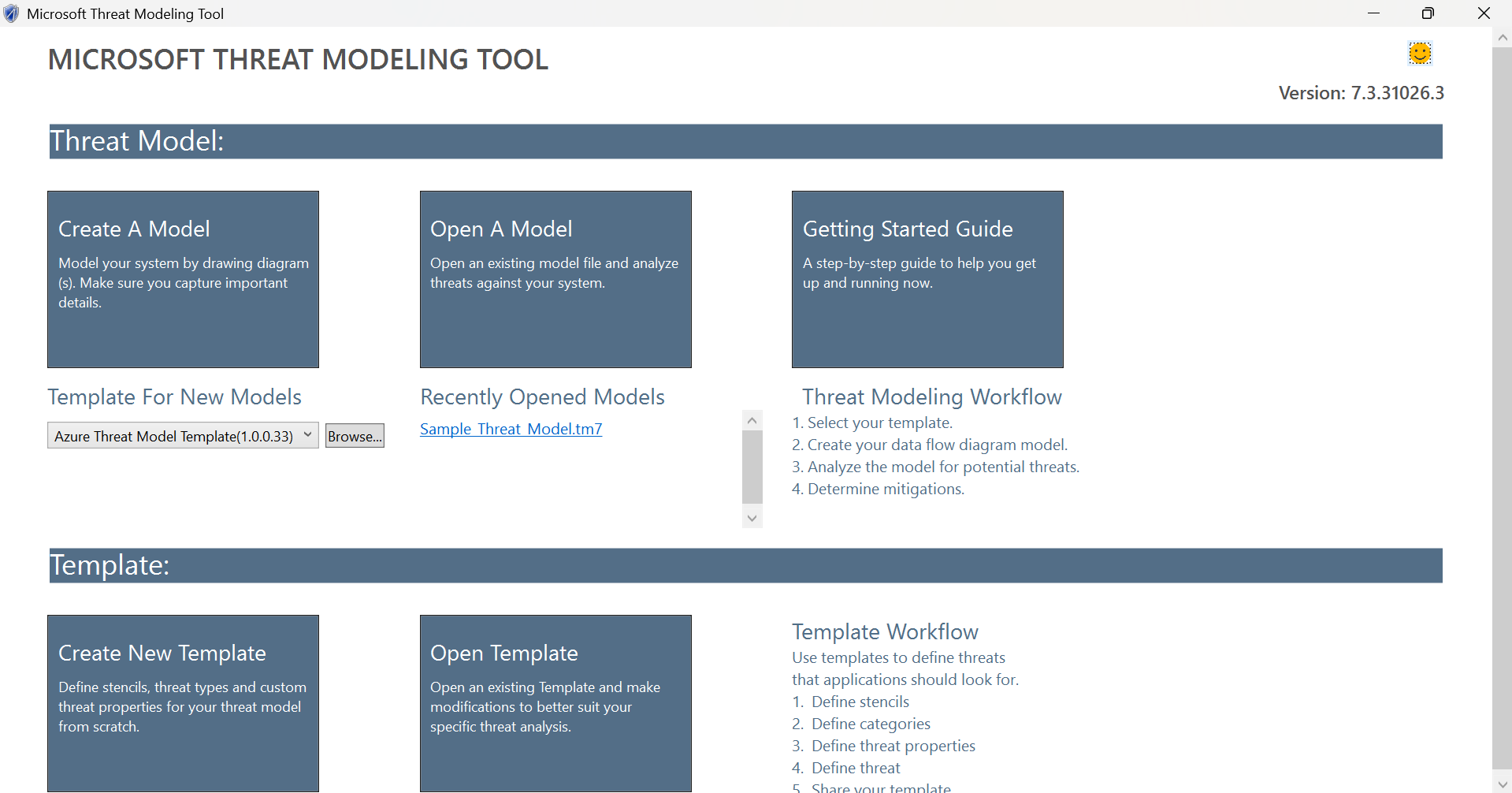
**Introduction**

**Threat modeling** is a proactive security practice that helps identify potential risks and vulnerabilities in software architecture during the design phase. Microsoft Threat Modeling Tool provides a graphical interface to model a system using **Data Flow Diagrams (DFDs)** and **automatically generates threat insights** using the **STRIDE** methodology.

**Procedure**

**Step 1: Tool Setup and Model Creation**

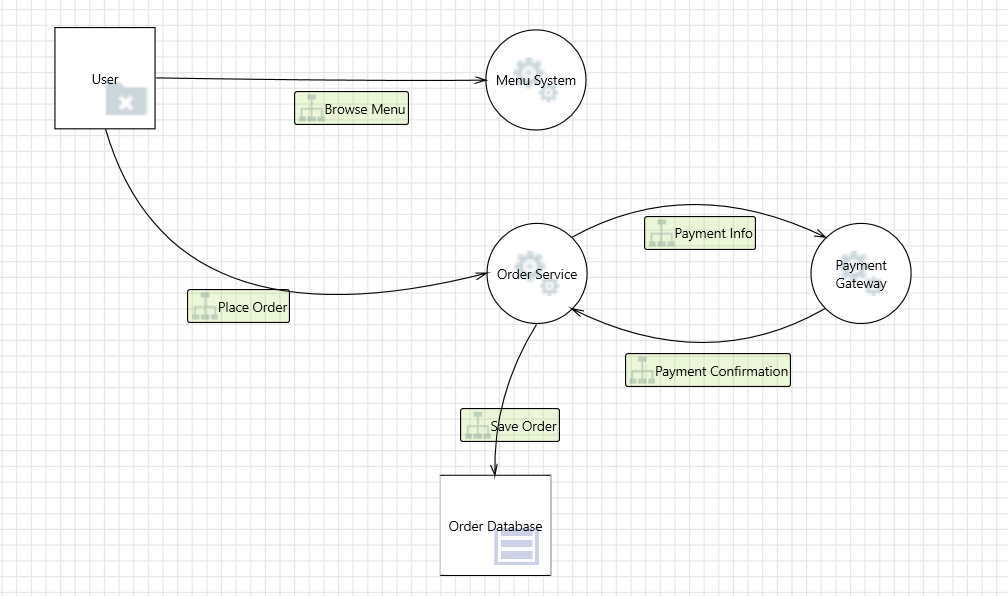
The Microsoft Threat Modeling Tool was installed and launched on a Windows system. A new model was created for an **Online Food Ordering System**.



**Step 2: Define System Architecture**

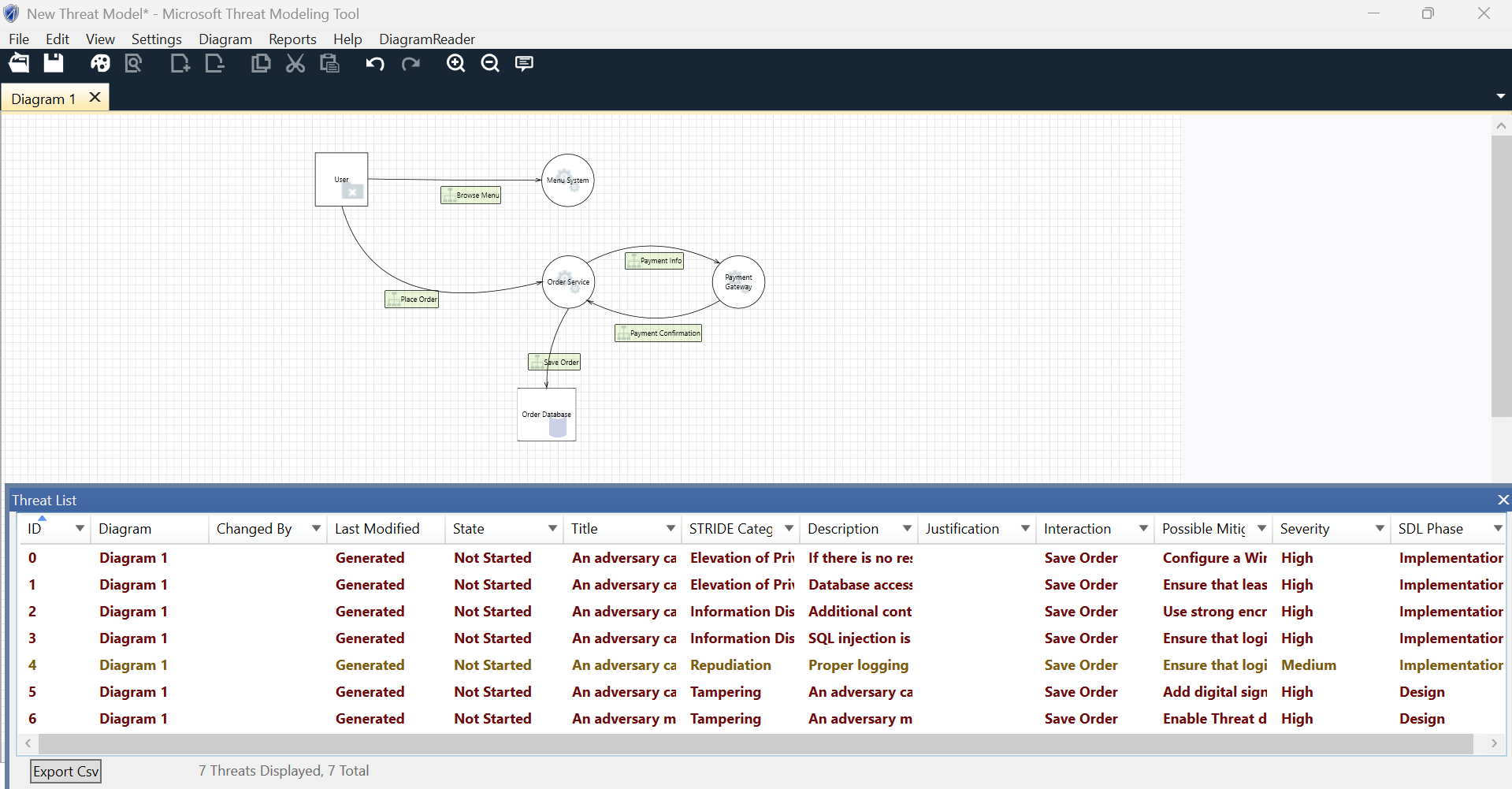
The architecture of the system was modeled using a Data Flow Diagram (DFD). The following components were added:

* **External Interactor:** User
* **Processes:** Menu System, Order Service, Payment Gateway
* **Data Store:** Order Database
* **Data Flows:**
  + User to Menu System (request/view food)
  + User to Order Service (submit order)
  + Order Service to Payment Gateway (payment request)
  + Order Service to Order Database (store order history)



**Step 3: Generate and Review Threats**

Using the built-in STRIDE model, Microsoft Threat Modeling Tool automatically identified potential threats related to each component and data flow. Each threat was categorized, described and reviewed for possible mitigation strategies.

****

**Threats Identified**

| **Component** | **Threat Type** | **Description** | **Suggested Mitigation** |
| --- | --- | --- | --- |
| **Tampering** | Order data can be modified during transmission | Data flow: User → Order Service | Use HTTPS and message integrity checks |
| **Information Disclosure** | Payment information may be exposed to third parties | Data flow: Order Service → Payment Gateway | Encrypt sensitive data during transmission |
| **Elevation of Privilege** | Attacker may gain access to admin functionalities | Order Service | Apply role-based access control (RBAC) |
| **Repudiation** | User may deny placing an order without evidence | Order Database | Implement proper logging and audit trails |

## 

## **Mitigation Recommendations**

## To address the identified threats in the Online Food Ordering System, the following security measures are recommended:

* Secure all communications with TLS/SSL to prevent data tampering or exposure
* Use encrypted storage for sensitive information in databases
* Implement access controls to prevent privilege escalation
* Enable activity logging for sensitive operations to support auditability
* Sanitize and validate all inputs from users to prevent malicious data injection