Technical Solution Design Document – CardioTriage-AI

# 1. Introduction

### 1.1 Purpose of the Document

This document outlines the design, configuration, and implementation of the CardioTriage AI, an AI-powered triage system aimed at automating the patient prioritization process within cardiology departments. The goal of this solution is to improve patient outcomes by ensuring that critical cases are handled promptly and efficiently.

### 1.2 Scope

This document covers the technical aspects of the system’s design, architecture, technology stack, security configurations, and integration with external tools and services. It provides a clear blueprint for the development and deployment of the solution.

### 1.3 Business Problem

Cardiology departments often struggle with prioritizing patients based on the severity of their conditions, which can lead to delays in treatment and affect patient outcomes. Critical cases, such as heart attacks, require immediate attention to prevent serious harm or even fatalities, while non-critical conditions, like stable angina, can overwhelm resources if not managed efficiently. The current manual process of triaging patients and scheduling appointments is time-consuming, prone to errors, and not always effective in ensuring that urgent cases are prioritized.

To address these challenges, there is a need of an AI-powered triage system that can automatically assess the severity of patient conditions, prioritize critical cases, streamline doctor scheduling, and optimize resource allocation, ensuring that life-threatening conditions are treated promptly while non-critical cases are managed efficiently.

### 1.4 Proposed Solution

This CardioTriage AI, an AI-powered triage system streamlines the cardiology department's workflow by automating patient prioritization and scheduling. Critical cases receive immediate attention, while non-critical cases are managed efficiently, leading to better resource utilization and improved patient care. The system also reduces the cognitive load on healthcare professionals by automating repetitive tasks, allowing them to focus on delivering high-quality care

# 2. Solution Architecture

### 2.1 Architecture Overview

CardioTriage AI is designed with Power Apps as the front-end interface, integrated with AI Builder & Copilot Custom Agent, while Dataverse serves as the backend database. Microsoft Bookings is utilized for managing scheduling, and Power Automate is employed to streamline the end-to-end automation processes. The application seamlessly connects to the database through native Power Apps connectors, ensuring smooth and efficient data accessibility.

As an additional security layer, all access keys are meticulously managed and stored within a dedicated Key Vault. This comprehensive approach guarantees the safeguarding of sensitive components and data within the CardioTriage AI Azure ecosystem.

### 2.2 High-Level Solution Architecture

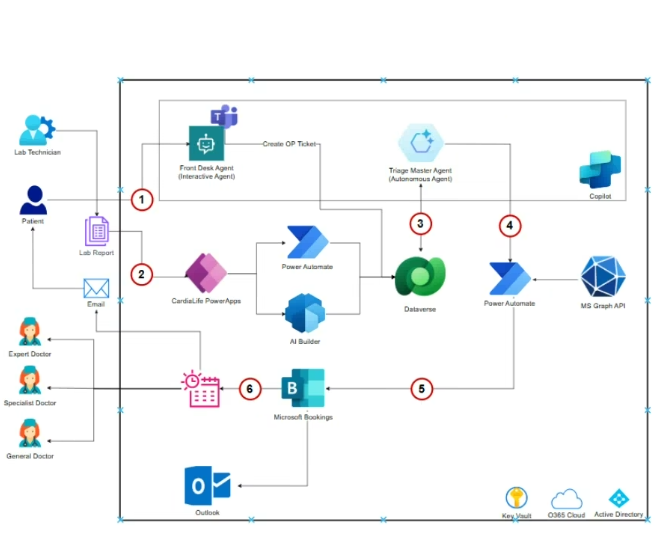


Figure 1: High-Level Solution Architecture Diagram

**User Interface (Front-End):**

* Power Apps: Serves as the primary interface for various users (Lab Technician, Front Desk Agent, Doctors). It provides a user-friendly way to interact with the system.
* Front Desk Agent (Interactive Agent): It serves as an interactive agent that helps with patient check-in and initial information gathering.
* TriageMaster (Autonomous Agent): This is the core AI-powered component, built using AI Builder and Copilot Custom Agent, that analyzes patient data to determine severity and prioritize cases based on guidelines.

**AI and Data Processing:**

* AI Builder: Used to build and train AI models for triage, likely utilizing machine learning to analyze patient symptoms, medical history, and lab results.
* Copilot Custom Agent: An extension of Microsoft Copilot, customized to fit the specific needs of the cardiology department. It provides conversational AI capabilities for assisting users and automating tasks.
* Dataverse: Acts as the central database, storing patient information, medical records, lab results, and scheduling data.

**Automation and Integration:**

* Power Automate: Automates workflows, such as sending notifications, scheduling appointments, and updating patient records.
* MS Graph API: Enables integration with Microsoft 365 services, such as Outlook for email notifications and Microsoft Bookings for scheduling.
* Microsoft Bookings: Manages appointment scheduling for doctors and patients.
* Email (Outlook): Used for communication and notifications.

**Security and Infrastructure:**

* Azure: The underlying cloud platform hosting all the components.
* Private Endpoints: Secure the Azure resources by ensuring all traffic stays within the Microsoft backbone network.
* Key Vault: Securely stores access keys and other sensitive information.
* O365 Cloud: Provides access to Microsoft 365 services like Outlook and Active Directory.
* Active Directory: Manages user authentication and authorization.

# 3. Technology Stack

The technology stack of the CardioTriage AI includes the following tools and platforms:

**Microsoft Power Platform:**

* Power Apps: Power Apps is a user-friendly platform that allows anyone to build custom apps with minimal coding. It helps create intuitive, front-end interfaces that connect to various data sources. In a solution design, Power Apps can be used to create easy-to-use applications for tasks like managing patient data or interacting with healthcare systems, all without requiring complex development skills.   
  Documentation Reference: [Official Microsoft Power Apps documentation - Power Apps | Microsoft Learn](https://learn.microsoft.com/en-us/power-apps/)
* Power Automate: Power Automate is a tool that helps automate repetitive tasks and workflows. It connects different services and apps to perform tasks automatically, saving time and reducing the chance of human error. In a technical solution, Power Automate can be used to streamline processes like scheduling patient appointments or syncing data across systems, making operations more efficient.  
  Documentation Reference: [Official Microsoft Power Automate documentation - Power Automate | Microsoft Learn](https://learn.microsoft.com/en-us/power-automate/)
* Dataverse: Dataverse is a secure, scalable database solution that stores and organizes data from various sources. It’s designed to handle large amounts of structured and unstructured data, making it easy to store and retrieve information in a consistent way. For a healthcare solution, Dataverse would store patient records, appointment data, and other critical information, ensuring data is secure and easily accessible.  
  Documentation Reference: [Dataverse documentation - Power Apps | Microsoft Learn](https://learn.microsoft.com/en-us/power-apps/maker/data-platform/)
* AI Builders: AI Builder is a tool that lets you add artificial intelligence to your apps without needing deep technical knowledge. It can help build features like predictive analytics or automated insights. In a healthcare solution, AI Builders could assist in assessing patient conditions, predicting future health risks, or helping staff make data-driven decisionsquickly.  
  Documentation Reference: [AI Builder documentation | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/)

**Copilot Agents:**

Copilot Agents are AI-powered assistants that help automate decision-making and assist with tasks. For healthcare, these agents analyze patient data to provide real-time suggestions or assessments, improving decision-making accuracy. Copilot Agents help clinicians by offering AI-driven recommendations based on patient data, making appointment processes smoother and more efficient.

Documentation Reference: [Copilots and generative AI in Power Platform - Power Platform | Microsoft Learn](https://learn.microsoft.com/en-us/power-platform/copilot)

**Microsoft Bookings:**

Microsoft Bookings is a scheduling tool that makes it easier for patients to book appointments and for staff to manage their schedules. It helps optimize resources by providing an easy-to-use interface for managing appointments. In a technical solution, Microsoft Bookings would simplify appointment management, allowing patients to schedule visits and staff to organize their time without any hassle.

Documentation Reference: [Microsoft Bookings | Microsoft Learn](https://learn.microsoft.com/en-us/microsoft-365/bookings/?view=o365-worldwide)

**Microsoft Graph:**

Microsoft Graph is an API that gives access to a wide range of data from Microsoft 365 services, like calendars, emails, and tasks. Microsoft Graph is being used to pull in important data from services like patient schedules or staff calendars, helping keep everything in sync and up to date in real-time.

Documentation Reference: [Microsoft Graph Fundamentals - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/paths/m365-msgraph-fundamentals/)

**Azure App Registrations:**

Azure App Registrations allows you to register apps within Azure Active Directory (Azure AD) for secure access and authentication. It ensures that only authorized users or applications can access sensitive data. Azure App Registrations help manage app security and ensure safe, seamless integration with services like Microsoft Graph or Power Apps.

Documentation Reference: [Implement app registration - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/implement-app-registration/)

**Azure Key Vault:**

Azure Key Vault is a service that stores sensitive data, such as passwords, API keys, and encryption keys, securely. It helps keep your most important information safe while allowing authorized apps or users to access it when needed. Azure Key Vault is being used to securely store secrets, such as database passwords or connection strings, ensuring everything is protected and accessible only by the right people or systems.

Documentation Reference: [Azure Key Vault documentation | Microsoft Learn](https://learn.microsoft.com/en-us/azure/key-vault/)

# 4. Configuration

To ensure a seamless and secure implementation of the solution, the following configuration details outline the necessary settings, integrations required for optimal performance and security-

### 4.1 Knowledge/Guidelines

Below are sample guideline documents we've used for customizing the knowledge base of the Custom Agent. Please feel free to use and refine them as needed for your specific requirements. Also Please note that these may not be 100% accurate.

1. Intelligent Cardiac Triage System Documentation:   
   Sample guideline documents used to customize the knowledge base for the Copilot Agent. These guidelines help in performing the criticality assessment.  
     
   **Intelligent Clinic Scheduling Guidelines**  
   *For GenAI Agents and Human Operators*
2. **Introduction**  
   This document serves as a comprehensive guide for AI agents and human operators to manage clinic appointments. It defines:  
   • Input requirements for booking.  
   • Logic for prioritizing appointments.  
   • Output formats for integration with downstream systems.  
   • Edge cases and validation rules.
3. **Input Definitions**

2.1 **Patient Booking Request**  
**Purpose:** Capture the patient’s required consultation time and urgency.  
**Structure:**

{

"TimeNeeded": 60, // \*Required\*. Consultation time (30 or 60 minutes).

"deadlineETA": "26-10-2023 10:30" // \*Required\*. Latest allowable booking time.

}

**Human Explanation:**  
• **TimeNeeded:** The duration the patient needs with the doctor. Only 30 or 60 minutes are allowed to align with clinic slots.  
• **deadlineETA:** The patient’s condition requires they be seen by this time. Example: A chest pain patient needing urgent care within 2 hours.

1. **Clinic Slot Structure**  
   **Purpose:** Define clinic hours and slot intervals.  
   **Details:**  
   • **Operating Hours:** 9:00 AM – 12:00 PM (6 slots/day).  
   • **Slot Times:**

* 09:00–09:30, 09:30–10:00, 10:00–10:30, 10:30–11:00, 11:00–11:30, 11:30–12:00.  
  **Human Explanation:**  
  • The clinic operates in 30-minute intervals.  
  • A 60-minute appointment requires two consecutive slots (e.g., 10:00–11:00).

1. **Scheduling Logic**  
   **Purpose:** Determine how to book appointments.

4.1 **Calculate Required Slots**  
**Rule:**

Slots Needed = TimeNeeded / 30

*Example:*  
• TimeNeeded = 60 → 2 slots (e.g., 10:00–11:00).  
**Human Explanation:**  
• The system converts time into slot units to ensure alignment with clinic hours.

4.2 **Slot Assignment Priority**  
**Rules:**

1. **Urgency:** Patients closer to their deadlineETA are prioritized.
2. **Criticality:** If deadlines are equal, prioritize High > Medium > Low criticality.  
   **Human Explanation:**  
   • A patient with a deadlineETA of today will be prioritized over one due tomorrow.  
   • Criticality ensures life-threatening cases are addressed first.
3. **Output Format**  
   **Purpose:** Provide machine-readable results for integration with booking systems.

5.1 **Success**  
If a slot is available and no conflicting appointment exists, the response should be:

{

"NewBookin

g": {

"ScheduledStartDateTime": "25-10-2023 10:30",

"ScheduledEndDateTime": "25-10-2023 11:30"

}

}

**Human Explanation:**  
• The patient is booked directly into the available slot without affecting any other appointment.

5.2 **Failure**  
If there is already an appointment present (i.e., no slots are available), the system should return:

{"No slots available"}

**Human Explanation:**  
• This response indicates that no valid slots exist for the requested time, and no rescheduling will be attempted.

1. **Edge Cases**  
   6.1 **TimeNeeded Exceeds 60 Minutes**  
   **Rule:** Reject requests with TimeNeeded > 60.  
   **Human Explanation:**  
   • Prevents overloading the clinic’s capacity.

6.2 **No Available Slots**  
**Resolution:** Return the failure JSON response.  
**Human Explanation:**  
• If an appointment already exists or no valid slot is free, the system responds with {"No slots available"}.

1. **Validation Rules**  
   **Purpose:** Ensure data integrity.
2. **TimeNeeded:** Must be 30 or 60 minutes.
3. **Slot Alignment:** Start/End times must match clinic slots.
4. **Deadlines:** Bookings must not exceed the provided deadlineETA.  
   **Human Explanation:**  
   • Prevents invalid bookings (e.g., a slot outside clinic hours).
5. **Appendices**

8.1 **Slot Matrix**

| **Date** | **Slot 1** | **Slot 2** | **Slot 3** | **Slot 4** | **Slot 5** | **Slot 6** |
| --- | --- | --- | --- | --- | --- | --- |
| 25-10-2023 | 9:00 | 9:30 | 10:00 | 10:30 | 11:00 | 11:30 |

**Human Explanation:**  
• A visual reference for clinic staff to verify slot availability.

8.2 **Sample Input/Output**

**Input:**

{

"TimeNeeded": 60,

"deadlineETA": "26-10-2023 10:30"

}

**Output (Success Scenario):**

{

"NewBooking": {

"ScheduledStartDateTime": "25-10-2023 10:00",

"ScheduledEndDateTime": "25-10-2023 11:00"

}

}

**Output (Failure Scenario):**

{"No slots available"}

**Human Explanation:**  
• In the success case, the patient is booked into two consecutive slots.  
• In the failure case, the system indicates that the requested slot is already taken.

1. Intelligent Clinic Scheduling Guidelines**:** Sample guideline documents used as knowledge for booking appointments. These guidelines outline the best practices and steps for scheduling appointments efficiently.

**Intelligent Clinic Scheduling Guidelines**  
**Version 2.2**  
*For GenAI Agents and Human Operators*

1. **Introduction**  
   This document serves as a comprehensive guide for AI agents and human operators to manage clinic appointments. It defines:  
   • Input requirements for booking.  
   • Logic for prioritizing appointments.  
   • Output formats for integration with downstream systems.  
   • Edge cases and validation rules.
2. **Input Definitions**

2.1 **Patient Booking Request**  
**Purpose:** Capture the patient’s required consultation time and urgency.  
**Structure:**

{

"TimeNeeded": 60, // \*Required\*. Consultation time (30 or 60 minutes).

"deadlineETA": "26-10-2023 10:30" // \*Required\*. Latest allowable booking time.

}

**Human Explanation:**  
• **TimeNeeded:** The duration the patient needs with the doctor. Only 30 or 60 minutes are allowed to align with clinic slots.  
• **deadlineETA:** The patient’s condition requires they be seen by this time. Example: A chest pain patient needing urgent care within 2 hours.

1. **Clinic Slot Structure**  
   **Purpose:** Define clinic hours and slot intervals.  
   **Details:**  
   • **Operating Hours:** 9:00 AM – 12:00 PM (6 slots/day).  
   • **Slot Times:**

* 09:00–09:30, 09:30–10:00, 10:00–10:30, 10:30–11:00, 11:00–11:30, 11:30–12:00.  
  **Human Explanation:**  
  • The clinic operates in 30-minute intervals.  
  • A 60-minute appointment requires two consecutive slots (e.g., 10:00–11:00).

1. **Scheduling Logic**  
   **Purpose:** Determine how to book appointments.

4.1 **Calculate Required Slots**  
**Rule:**

Slots Needed = TimeNeeded / 30

*Example:*  
• TimeNeeded = 60 → 2 slots (e.g., 10:00–11:00).  
**Human Explanation:**  
• The system converts time into slot units to ensure alignment with clinic hours.

4.2 **Slot Assignment Priority**  
**Rules:**

1. **Urgency:** Patients closer to their deadlineETA are prioritized.
2. **Criticality:** If deadlines are equal, prioritize High > Medium > Low criticality.  
   **Human Explanation:**  
   • A patient with a deadlineETA of today will be prioritized over one due tomorrow.  
   • Criticality ensures life-threatening cases are addressed first.
3. **Output Format**  
   **Purpose:** Provide machine-readable results for integration with booking systems.

5.1 **Success**  
If a slot is available and no conflicting appointment exists, the response should be:

{

"NewBooking": {

"ScheduledStartDateTime": "25-10-2023 10:30",

"ScheduledEndDateTime": "25-10-2023 11:30"

}

}

**Human Explanation:**  
• The patient is booked directly into the available slot without affecting any other appointment.

5.2 **Failure**  
If there is already an appointment present (i.e., no slots are available), the system should return:

{"No slots available"}

**Human Explanation:**  
• This response indicates that no valid slots exist for the requested time, and no rescheduling will be attempted.

1. **Edge Cases**  
   6.1 **TimeNeeded Exceeds 60 Minutes**  
   **Rule:** Reject requests with TimeNeeded > 60.  
   **Human Explanation:**  
   • Prevents overloading the clinic’s capacity.

6.2 **No Available Slots**  
**Resolution:** Return the failure JSON response.  
**Human Explanation:**  
• If an appointment already exists or no valid slot is free, the system responds with {"No slots available"}.

1. **Validation Rules**  
   **Purpose:** Ensure data integrity.
2. **TimeNeeded:** Must be 30 or 60 minutes.
3. **Slot Alignment:** Start/End times must match clinic slots.
4. **Deadlines:** Bookings must not exceed the provided deadlineETA.  
   **Human Explanation:**  
   • Prevents invalid bookings (e.g., a slot outside clinic hours).
5. **Appendices**

8.1 **Slot Matrix**

| **Date** | **Slot 1** | **Slot 2** | **Slot 3** | **Slot 4** | **Slot 5** | **Slot 6** |
| --- | --- | --- | --- | --- | --- | --- |
| 25-10-2023 | 9:00 | 9:30 | 10:00 | 10:30 | 11:00 | 11:30 |

**Human Explanation:**  
• A visual reference for clinic staff to verify slot availability.

8.2 **Sample Input/Output**

**Input:**

{

"TimeNeeded": 60,

"deadlineETA": "26-10-2023 10:30"

}

**Output (Success Scenario):**

{

"NewBooking": {

"ScheduledStartDateTime": "25-10-2023 10:00",

"ScheduledEndDateTime": "25-10-2023 11:00"

}

}

**Output (Failure Scenario):**

{"No slots available"}

**Human Explanation:**  
• In the success case, the patient is booked into two consecutive slots.  
• In the failure case, the system indicates that the requested slot is already taken.

### 4.2 MS Bookings

We have leveraged Microsoft Azure App registrations to integrate the Microsoft Graph API, enabling us to generate access tokens. These tokens are used to perform various operations related to scheduling and managing appointments in Microsoft Bookings through the REST API.

# 5. Usage

### 5.1 User Roles and Permissions

Different user roles within the application, such as healthcare professionals, administrators, and technical support, will have varying levels of access to the system:

Healthcare Professionals: Can input patient data, view triage priorities, and schedule appointments.

Administrators: Manage users, workflows, and security settings.

Technical Support: Monitor the system and handle troubleshooting.

5.2 Workflow and Process Overview  
  
Below are the processes that make up the solution, ensuring a comprehensive and efficient workflow for healthcare delivery

* Patient Check-In: The AI agent registers the patient at the front desk and generates an OP ticket for further processing.
* Report Upload & Processing: A lab assistant uploads test reports to PowerApps, where AI Builder reads the data, processes it, and updates Dataverse.
* Triage Activation: The TriageMaster AI is automatically triggered when a lab report is submitted, initiating the evaluation process.
* Data Evaluation: AI analyzes report values to determine the required doctor expertise, assess case urgency, and recommend consultation if needed.
* Appointment Scheduling: If a consultation is required, AI finds the right doctor and reserves an available slot in Microsoft Bookings.
* Instant Notification: Patients receive immediate email alerts confirming their appointment or providing health tips when a consultation is not needed.

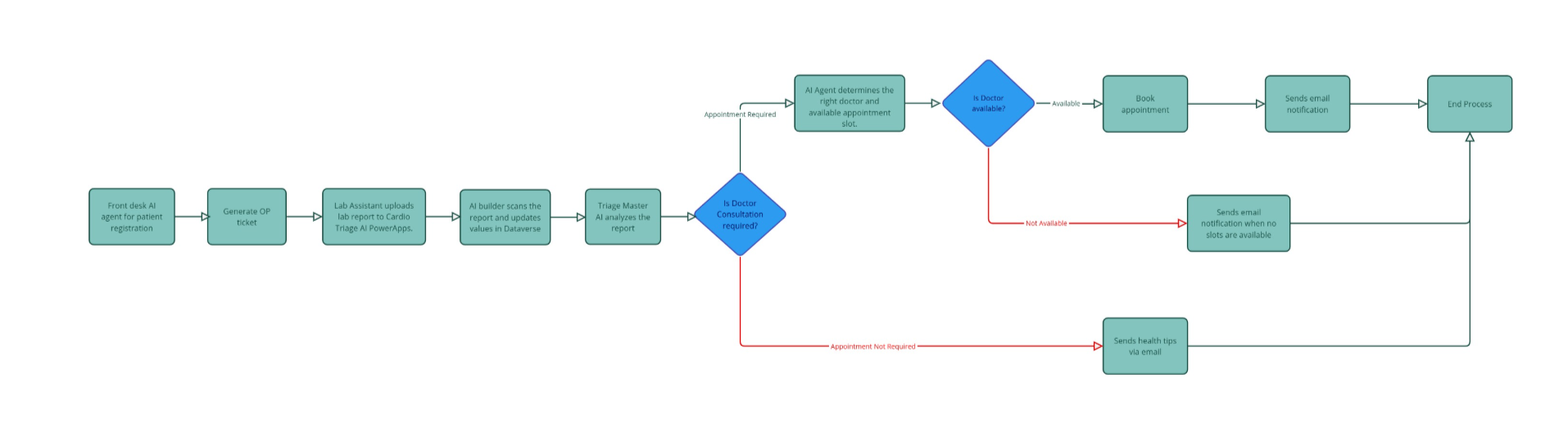


Figure 2: Process Flow Diagram

# 6. Security and Compliance

### 6.1 Data Security

The system ensures data security using:

Private Endpoints to isolate all communications within the secure Azure network.

Azure Key Vault to securely store credentials and sensitive information.

Data Encryption: All sensitive patient data is encrypted both in transit and at rest.

### 6.2 Compliance

The solution adheres to industry standards and regulations, including HIPAA and GDPR, for healthcare and patient data protection.

# 7. Licenses Requirement

* Microsoft Bookings: Microsoft 365 E3 or E5, Microsoft 365 Business Standard or higher plans.
* Copilot Studio: Microsoft 365 Copilot access, which is available for Enterprise E3/E5.
* AI Builder: Power Apps Premium or Power Automate Premium licenses and AI Builder add-ons.
* Power Platform: Power Apps or Power Automate Premium licenses, with potential add-ons for more advanced features

# 8. Limitations & Known Issues

* Microsoft Bookings is currently in the Preview stage, so it's not yet recommended for full-scale production use.
* Microsoft Bookings doesn’t have a native Power Automate connector for creating appointments. Instead, the current version relies on the Microsoft Graph API to manage appointments.
* Microsoft Bookings doesn’t automatically sync appointment data with Dataverse. Unlike other Microsoft apps, you’ll need to set up custom workflows manually to store and manage this information.

# 9. Conclusion

The CardioTriage AI solution is designed to revolutionize patient triage in cardiology by automating prioritization and scheduling, ensuring critical cases receive immediate attention. Leveraging AI-powered decision-making, seamless automation, and secure cloud infrastructure, this system enhances efficiency, reduces delays, and optimizes resource allocation.

By integrating Power Apps, AI Builder, Copilot Custom Agents, Dataverse, and Microsoft Bookings, the solution streamlines workflows while maintaining strict security standards. The result is a smarter, faster, and more reliable approach to managing patient care, ultimately improving outcomes and reducing the burden on healthcare professionals.