# System Interaction Diagrams for AcmePlex Movie Theater Reservation System

This document outlines the steps to create system interaction diagrams for four major use cases in the AcmePlex Movie Theater Reservation System. Each use case demonstrates how system components interact to achieve specific functionality.

## Four Major Use Cases

1. Browse and Select a Movie: Searching, filtering, and selecting movies.  
2. Book a Ticket: Seat selection, checking availability, and confirming the booking.  
3. Make Payment: Entering payment details, verifying payment, and sending confirmation.  
4. Send Email Notification: Sending email confirmations to users after a booking or payment.

## Diagram 1: Browse and Select a Movie

\*\*Participants:\*\*  
- User  
- System  
- Database  
  
\*\*Flow:\*\*  
1. User → System: 'Search for a movie.'  
2. System → Database: 'Query movies based on user input.'  
3. Database → System: 'Return matching movies.'  
4. System → User: 'Display search results.'  
  
\*\*Diagram Type:\*\* Sequence Diagram.

## Diagram 2: Book a Ticket

\*\*Participants:\*\*  
- User  
- System  
- Database  
  
\*\*Flow:\*\*  
1. User → System: 'Select seat for booking.'  
2. System → Database: 'Check seat availability.'  
3. Database → System: 'Confirm seat is available.'  
4. User → System: 'Confirm booking.'  
5. System → Database: 'Update seat status to booked.'  
  
\*\*Diagram Type:\*\* Sequence Diagram.

## Diagram 3: Make Payment

\*\*Participants:\*\*  
- User  
- System  
- Payment Gateway  
  
\*\*Flow:\*\*  
1. User → System: 'Enter payment details.'  
2. System → Payment Gateway: 'Submit payment request.'  
3. Payment Gateway → System: 'Return payment success/failure response.'  
4. System → User: 'Show payment confirmation or error.'  
  
\*\*Diagram Type:\*\* Sequence Diagram.

## Diagram 4: Send Email Notification

\*\*Participants:\*\*  
- User  
- System  
- Database  
- Email Service  
  
\*\*Flow:\*\*  
1. User → System: 'Confirm Booking or Payment.'  
2. System → Database: 'Fetch user email and booking details.'  
3. Database → System: 'Return email and booking details.'  
4. System → Email Service: 'Send email notification.'  
5. Email Service → User: 'Deliver email confirmation.'  
  
\*\*Diagram Type:\*\* Sequence or Collaboration Diagram.

## Responsibility Assignment

Each team member is responsible for one use case, and their name must appear on the related page:  
1. Browse and Select a Movie – [Team Member 1's Name]  
2. Book a Ticket – [Team Member 2's Name]  
3. Make Payment – [Team Member 3's Name]  
4. Send Email Notification – [Your Name or Team Member 4's Name]

## Steps to Create Diagrams in Draw.io

1. Open Draw.io and create a new blank diagram.  
   2. Add the participants (User, System, etc.) as rectangles.  
   3. Draw lifelines below each participant for sequence diagrams or relationships for collaboration diagrams.  
   4. Add arrows for messages, with clear labels (e.g., 'Search for Movie').  
   5. Use activation bars where appropriate to show processing steps.  
   6. Save and export each diagram as PNG, JPEG, or PDF.

# Step-by-Step Instructions to Create System Interaction Diagrams in Draw.io

This document provides detailed instructions on how to create system interaction diagrams in Draw.io for the AcmePlex Movie Theater Reservation System.

## General Steps for Creating Diagrams in Draw.io

1. Open Draw.io (https://app.diagrams.net/) and create a new blank diagram.  
2. Title your diagram (e.g., 'Sequence Diagram for Browse and Select a Movie').  
3. Use the left panel to drag shapes and objects onto the canvas.  
4. Align and connect the elements using arrows or lines to represent relationships or flows.  
5. Label all components and interactions for clarity.  
6. Use the export feature to save your diagram as PNG, JPEG, or PDF.

## 1. Sequence Diagram

\*\*Steps to Create a Sequence Diagram:\*\*  
1. Drag rectangles from the left panel to represent participants (e.g., User, System, Database).  
2. Add dashed lines below each rectangle to represent lifelines.  
3. Use arrows to indicate messages between participants:  
 - For example: 'User → System: Search Movie.'  
 - Label each arrow with the corresponding message.  
4. Add narrow rectangles over lifelines to represent activation bars where processing occurs.  
5. Ensure the message flow reflects the order of interactions.  
6. Save and export the diagram.

## Detailed Steps for Each Interaction Diagram

### Diagram 1: Browse and Select a Movie

1. Add participants: User, System, Database.  
2. Create the following interactions:  
 - User → System: 'Search for a movie.'  
 - System → Database: 'Query movies based on user input.'  
 - Database → System: 'Return matching movies.'  
 - System → User: 'Display search results.'  
3. Use dashed lifelines for participants and activation bars for active processes.

### Diagram 2: Book a Ticket

1. Add participants: User, System, Database.  
2. Create the following interactions:  
 - User → System: 'Select seat for booking.'  
 - System → Database: 'Check seat availability.'  
 - Database → System: 'Confirm seat is available.'  
 - User → System: 'Confirm booking.'  
 - System → Database: 'Update seat status to booked.'  
3. Use arrows to indicate the message flow.

### Diagram 3: Make Payment

1. Add participants: User, System, Payment Gateway.  
2. Create the following interactions:  
 - User → System: 'Enter payment details.'  
 - System → Payment Gateway: 'Submit payment request.'  
 - Payment Gateway → System: 'Return payment success/failure response.'  
 - System → User: 'Show payment confirmation or error.'  
3. Use activation bars for the payment process.

### Diagram 4: Send Email Notification

1. Add participants: User, System, Database, Email Service.  
2. Create the following interactions:  
 - User → System: 'Confirm Booking or Payment.'  
 - System → Database: 'Fetch user email and booking details.'  
 - Database → System: 'Return email and booking details.'  
 - System → Email Service: 'Send email notification.'  
 - Email Service → User: 'Deliver email confirmation.'  
3. Highlight the email service's role using arrows and activation bars.

## Final Notes

Ensure diagrams are labeled clearly and interactions are sequentially numbered where applicable. Use alignment tools in Draw.io for a clean layout. Save and export diagrams for submission.