

```
Import java.awt.*;
Import java.awt.event.*;
Import java.util.ArrayList;
Import java.util.HashMap;
```

```
Class HelpDeskSystem extends Frame {
```

```
    // Ticket class to manage ticket details
```

```
    Static class Ticket {
```

```
        String ticketID;
```

```
        String issue;
```

```
        String status;
```

```
        String solution;
```

```
        Ticket(String ticketID, String issue, String solution) {
```

```
            This.ticketID = ticketID;
```

```
            This.issue = issue;
```

```
            This.solution = solution;
```

```
            This.status = "Resolved";
```

```
        }
```

```
    }
```

```
    // User interface elements
```

```
    TextField userInput, ticketInput, searchInput;
```

```
    TextArea outputArea;
```

```
    ArrayList<Ticket> tickets = new ArrayList<>();
```

```
    HashMap<String, String> knowledgeBase = new HashMap<>();
```

```
    Int ticketCounter = 1;
```

```
    Public HelpDeskSystem() {
```

```
// Populate Knowledge Base with predefined issues and solutions

knowledgeBase.put("login issue", "Reset your password or contact support for account recovery.");

knowledgeBase.put("network issue", "Restart your router and check your internet settings.");

knowledgeBase.put("software crash", "Reinstall the software and update to the latest version.");

knowledgeBase.put("hardware failure", "Check device connections or contact hardware support.");

knowledgeBase.put("printer not working", "Ensure the printer is properly connected and has paper/ink.");

knowledgeBase.put("email not syncing", "Check email server settings and ensure the internet connection is stable.");

knowledgeBase.put("slow computer", "Close unnecessary programs, clear cache, or increase RAM.");

knowledgeBase.put("data loss", "Restore from a backup or use data recovery software.");

knowledgeBase.put("blue screen error", "Restart your system and check for driver updates or hardware issues.");

knowledgeBase.put("account locked", "Contact support to unlock your account after verification.");
```

```
// Set up the frame
```

```
setTitle("Help Desk Management System");
```

```
setSize(500, 600);
```

```
setLayout(new FlowLayout());
```

```
setVisible(true);
```

```
// Create input and output areas
```

```
Label userLabel = new Label("Enter User Role (Customer/Agent):");
```

```
userInput = new TextField(20);
```

```
Label ticketLabel = new Label("Describe Your Issue:");
```

```
ticketInput = new TextField(40);  
Button createTicketButton = new Button("Create Ticket");  
Label searchLabel = new Label("Search Ticket by ID:");  
searchInput = new TextField(20);  
Button searchTicketButton = new Button("Search Ticket");  
outputArea = new TextArea(20, 40);
```

```
// Add components to the frame
```

```
Add(userLabel);  
Add(userInput);  
Add(ticketLabel);  
Add(ticketInput);  
Add(createTicketButton);  
Add(searchLabel);  
Add(searchInput);  
Add(searchTicketButton);  
Add(outputArea);
```

```
// Event handling for creating tickets
```

```
createTicketButton.addActionListener(e -> createTicket());
```

```
// Event handling for searching tickets
```

```
searchTicketButton.addActionListener(e -> searchTicket());
```

```
// Close window event
```

```
addWindowListener(new WindowAdapter() {  
    public void windowClosing(WindowEvent e) {  
        dispose();  
    }  
})
```

```
});  
}
```

```
// Method to create a ticket
```

```
Private void createTicket() {
```

```
    String role = userInput.getText().trim();
```

```
    String issue = ticketInput.getText().trim().toLowerCase();
```

```
    If (role.isEmpty() || issue.isEmpty()) {
```

```
        outputArea.setText("Error: Role and issue cannot be empty.");
```

```
        return;
```

```
    }
```

```
    String ticketID = "TICKET" + (ticketCounter++);
```

```
    String solution = knowledgeBase.getDefault(issue, "No solution found.  
Please contact support.");
```

```
    Ticket newTicket = new Ticket(ticketID, issue, solution);
```

```
    Tickets.add(newTicket);
```

```
    outputArea.setText("Ticket Created Successfully!\n");
```

```
    outputArea.append("Ticket ID: " + ticketID + "\nRole: " + role + "\nIssue: " +  
issue + "\nSolution: " + solution + "\nStatus: " + newTicket.status + "\n");
```

```
    ticketInput.setText("");
```

```
}
```

```
// Method to search for a ticket
```

```
Private void searchTicket() {
```

```
    String ticketID = searchInput.getText().trim();
```

```
    If (ticketID.isEmpty()) {
```

```
        outputArea.setText("Error: Ticket ID cannot be empty.");  
        return;  
    }
```

```
    For (Ticket ticket : tickets) {  
        If (ticket.ticketID.equalsIgnoreCase(ticketID)) {  
            outputArea.setText("Ticket Found!\n");  
            outputArea.append("Ticket ID: " + ticket.ticketID + "\nIssue: " + ticket.issue  
+ "\nSolution: " + ticket.solution + "\nStatus: " + ticket.status + "\n");  
            return;  
        }  
    }  
    outputArea.setText("Error: Ticket not found.");  
}
```

```
Public static void main(String[] args) {  
    New HelpDeskSystem();  
}  
}
```