UNIVERSITI TEKNOLOGI PETRONAS

TFB1033/TEB1043: OBJECT ORIENTED PROGRAMMING

SEMESTER MAY 2025

PROJECT REPORT

TITLE: HR MANAGEMENT SYSTEM

LECTURER: DR NORDIN ZAKARIA

NAME	ID
Lee Jie Yie	22011597
Aminudin Razif Bin Arman	22007578
Muhammad Danish Bin Jamal	22011473
Syed Muhammad Kadzim Alattas Bin Syed Sheikh Alattas	22011248
Desmond Sua Anak Tony	24007648

TABLE OF CONTENT

1.0 DESCRIPTION

This project, Human Resources (HR) Management System is developed using C# programming language with Windows Form (WinForms) Framework. The system purpose is to automate essential HR task in a company. For example, such as employee data management, leave application tracking system, staff attendance tracking and payroll processing.

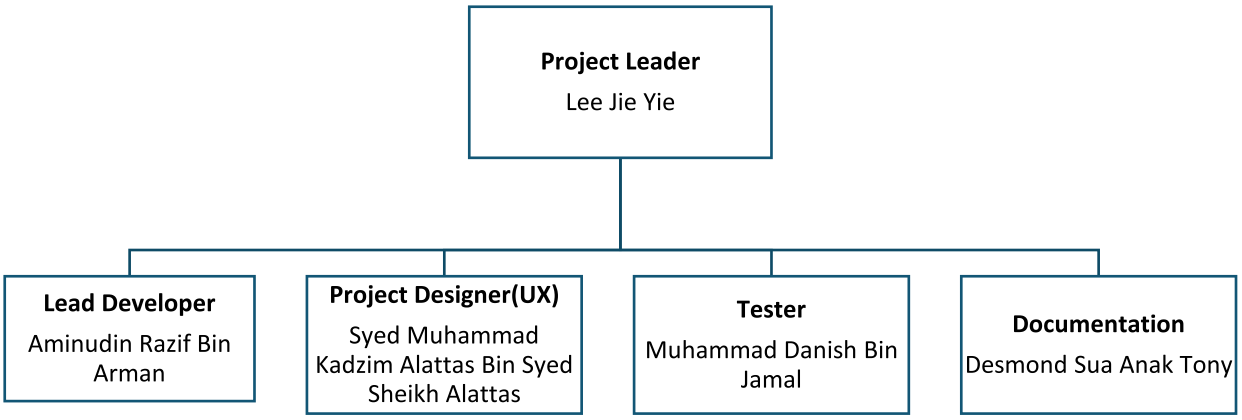
The application feature user friendly graphical user interface (GUI) that enables HR staff to perform their task efficiently. This also includes adding, updating, deleting and finding employee records. The reason is to ensure all employee related information is securely stored and easily accessible.

Key features of the system include:

- **Employee Management**: Maintain employee personal information such as contact details and job description
- **User Login Panel:** For HR personnel to access the mainframe of the program for their record.
- **Leave Management:** Tracking employee leave request, approval and leave balances.
- **Attendance Tracking:** Tracking employee attendance status.
- **Payroll Module System:** Initiate the salary of the employee by calculating hours worked, bonuses and penalty deduction.

To sum up, this system helps reduce manual workload, minimizes errors in record keeping, and improves overall HR process efficiency within an organization. It is suitable for small and medium enterprises (SME) seeking for a desktop-based HR management solution.

2.0 TEAM ORGANISATION



Description

1. Project Leader

Person: Lee Jie Yie

Role

- The main coordinator and overall decision-maker for the project.
- Responsible for planning, organizing, and ensuring the team meets project goals and deadlines.
- Communicates with stakeholders and resolves any major issues.

2. Lead Developer

Person: Aminudin Razif Bin Arman

Role:

- Responsible for coding, designing the software architecture, and solving technical problems.
- Guides other developers (if any) and ensures code quality.
- Implements core features according to the project requirements.

3. Project Designer (UX)

Person: Syed Muhammad Kadzim Alattas Bin Syed Sheikh Alattas

Role:

- Focuses on user experience (UX) and the design aspect of the project.
- Ensures the application is user-friendly and visually appealing.

4. Tester

Person: Muhammad Danish Bin Jamal

Role:

- Responsible for testing the software for bugs and errors.
- Develops test cases and scenarios.
- Ensures the software works as intended and meets quality standards.

5. Documentation

Person: Desmond Sua Anak Tony

Role:

- Prepares and maintains all project documentation
- Ensures records are up to date and comprehensive for future reference.
- Helps team members and users understand the system and processes.

3.0 UNIFIED MODELLING LANGUAGE(UML)

3.1 UML DIAGRAM

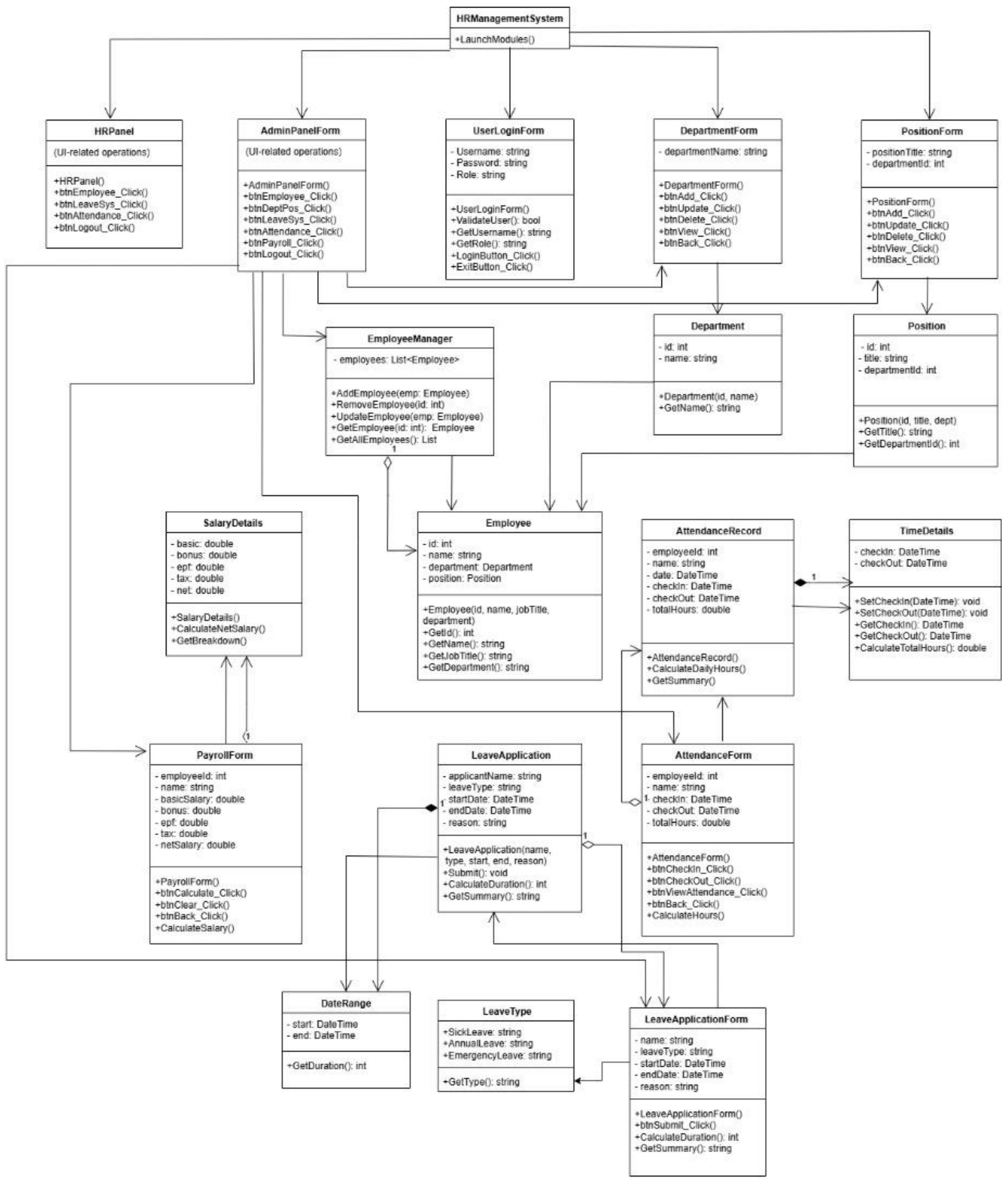


Figure 1: UML Diagram

3.2 UML DETAILS

1. Main Components & UI Classes

HRManagementSystem

- Method: LaunchModules()
- Role: Central system class, responsible for launching different modules/forms.

Panels and Forms (UI Layer)

- HRPanel / AdminPanelForm / UserLoginForm / DepartmentForm / PositionForm
    - Role: Handle UI-related operations (buttons and user actions).
    - Methods: Button click event handlers (e.g., btnEmployee_Click(), btnAdd_Click()).

2. Core Domain Classes

EmployeeManager

- Attributes: employees: List<Employee>
- Methods:
    - AddEmployee(emp: Employee)
    - RemoveEmployee(id: int)
    - UpdateEmployee(emp: Employee)
    - GetEmployee(id: int): Employee
    - GetAllEmployees(): List<Employee>
- Relationship: Aggregates/manages multiple Employee objects.

Employee

- Attributes:
    - id: int
    - name: string

  - department: Department
  - position: Position
- Methods:
  - GetId()
  - GetName()
  - GetJobTitle()
  - GetDepartment()
- Relationships:
  - Association with Department and Position
  - Linked to: SalaryDetails, AttendanceRecord, LeaveApplication

Department

- Attributes:
  - id: int
  - name: string
- Methods:
  - GetName()
  - GetId()
- Relationship: Used by Employee, referenced in DepartmentForm.

Position

- Attributes:
  - id: int
  - title: string
  - departmentId: int

- Methods:
  - GetId()
  - GetTitle()
  - GetDepartmentId()
- Relationship: Used by Employee, referenced in PositionForm.

3. Salary & Payroll

SalaryDetails

- Attributes:
  - basic, bonus, ot, net, total (all double)
- Methods:
  - CalculateNetSalary()
  - GetBreakdown()
- Relationship: Associated with PayrollForm and Employee.

PayrollForm

- Attributes: Employee details and salary breakdown.
- Methods: Button event handlers for payroll actions, salary calculation.
- Relationship: Uses SalaryDetails.

4. Attendance Management

AttendanceRecord

- Attributes:
  - employeeId: int
  - name: string
  - date: DateTime

    - checkIn: DateTime
    - checkOut: DateTime
    - totalHours: double
- Methods:
    - CalculateDailyHours()
    - GetSummary()
- Relationship:
    - Composition with TimeDetails (AttendanceRecord uses TimeDetails).

TimeDetails

- Attributes:
    - checkIn: DateTime
    - checkOut: DateTime
- Methods:
    - SetCheckIn(DateTime)
    - SetCheckOut(DateTime)
    - CalculateTotalHours()
- Relationship: Used by AttendanceRecord.

AttendanceForm

- Attributes: Attendance data.
- Methods: Button event handlers for attendance, CalculateHours().
- Relationship: Connects UI to AttendanceRecord.

5. Leave Management

LeaveApplication

- Attributes:
    - applicantName: string
    - leaveType: string
    - startDate: DateTime
    - endDate: DateTime
    - reason: string
- Methods:
    - CalculateDuration()
    - GetSummary()
- Relationship:
    - Association with LeaveType and DateRange.

DateRange

- Attributes:
    - start: DateTime
    - end: DateTime
- Methods:
    - GetDuration()
- Relationship: Used by LeaveApplication.

LeaveType

- Attributes:
    - SickLeave: string

    - AnnualLeave: string
    - EmergencyLeave: string
- Methods:
    - GetType()
- Relationship: Used by LeaveApplication.

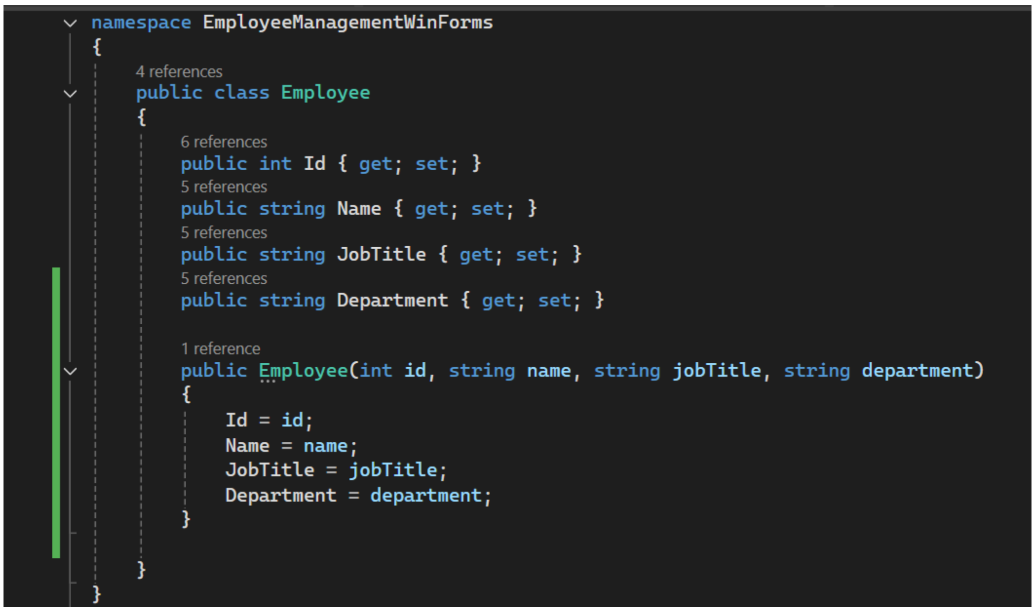
LeaveApplicationForm

- Attributes: All leave application fields.
- Methods:
    - btnSubmit_Click()
    - btnClear_Click()
    - btnBack_Click()
    - GetSummary()
- Relationship: Connects UI to LeaveApplication

4.0 IMPLEMENTATION

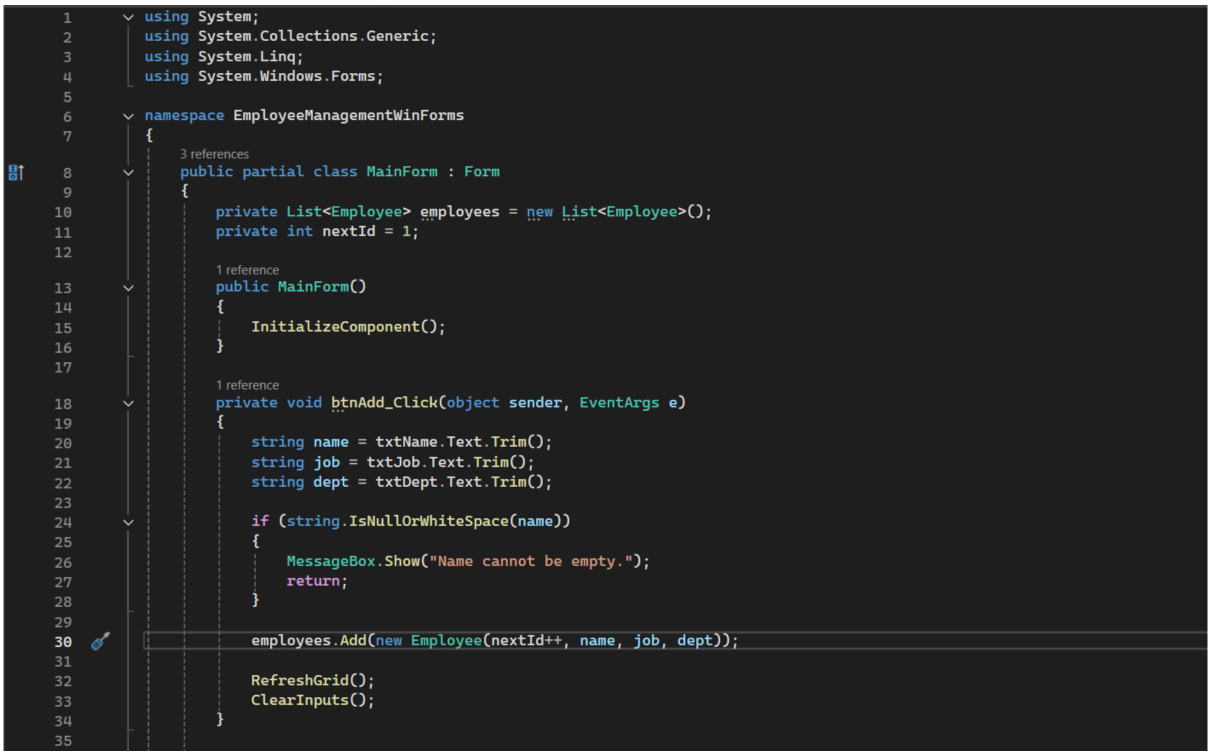
4.1 CODE

Employee Management



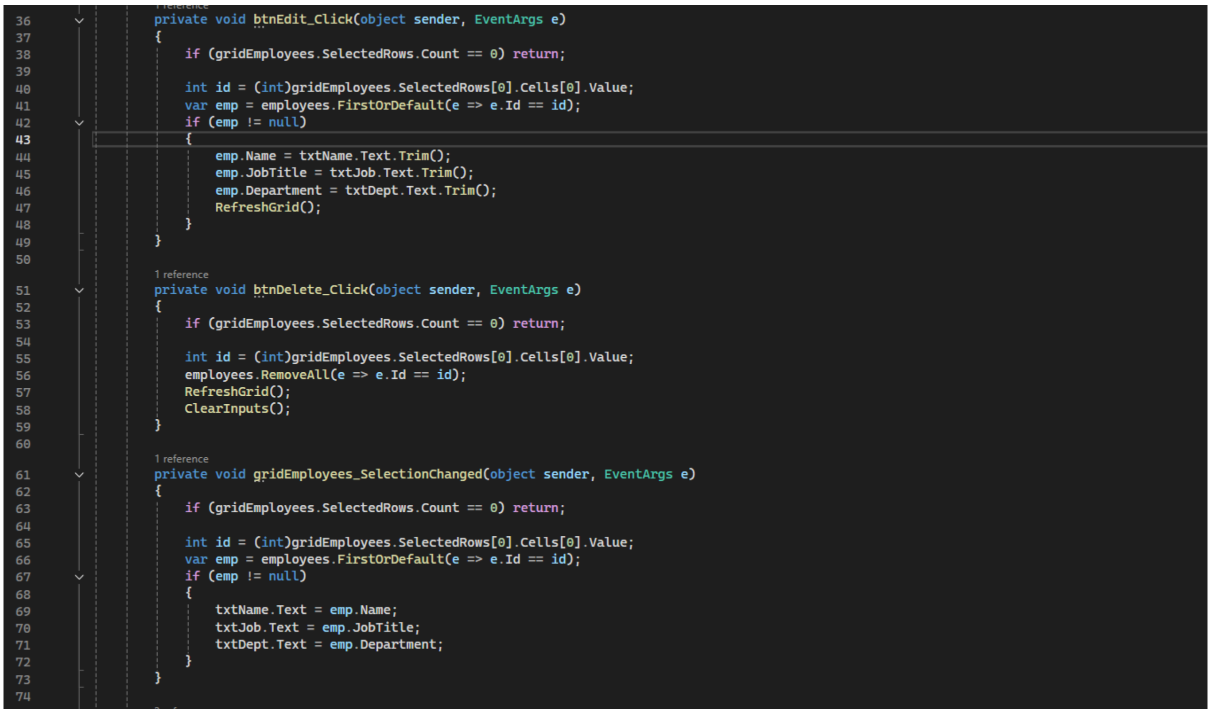
```csharp
namespace EmployeeManagementWinForms
{
    4 references
    public class Employee
    {
        6 references
        public int Id { get; set; }
        5 references
        public string Name { get; set; }
        5 references
        public string JobTitle { get; set; }
        5 references
        public string Department { get; set; }

        1 reference
        public Employee(int id, string name, string jobTitle, string department)
        {
            Id = id;
            Name = name;
            JobTitle = jobTitle;
            Department = department;
        }

    }
}
```

Figure 2: Employee.cs



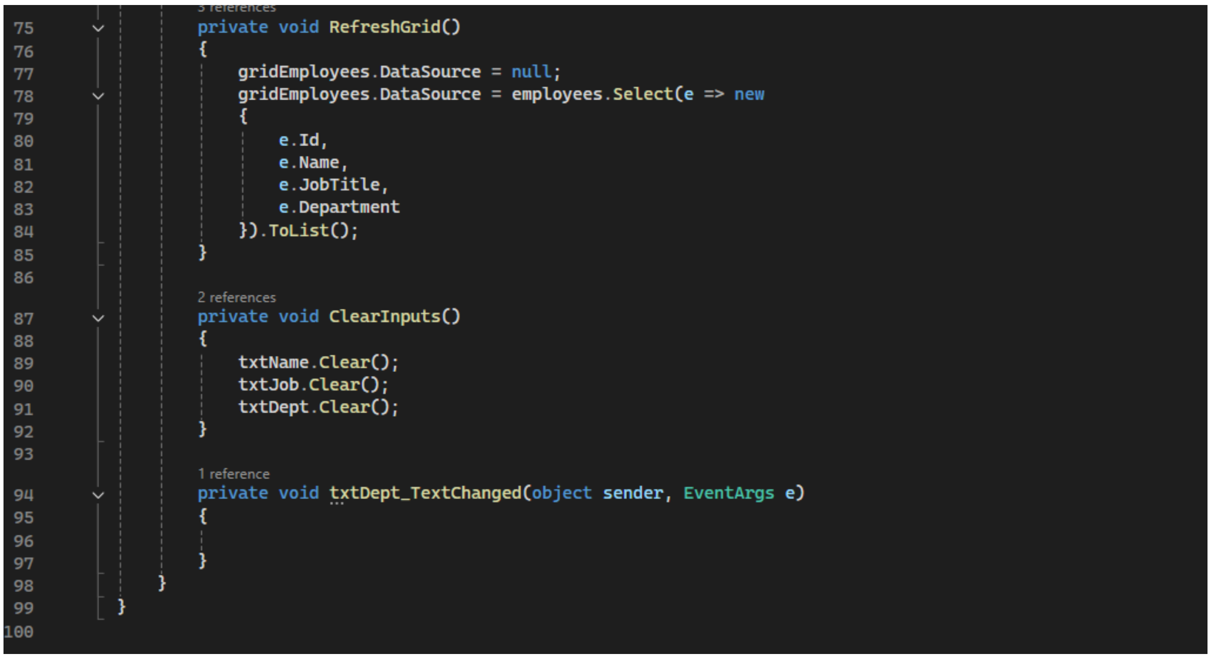
```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace EmployeeManagementWinForms
{
    3 references
    public partial class MainForm : Form
    {
        private List<Employee> employees = new List<Employee>();
        private int nextId = 1;

        1 reference
        public MainForm()
        {
            InitializeComponent();
        }

        1 reference
        private void btnAdd_Click(object sender, EventArgs e)
        {
            string name = txtName.Text.Trim();
            string job = txtJob.Text.Trim();
            string dept = txtDept.Text.Trim();

            if (string.IsNullOrWhiteSpace(name))
            {
                MessageBox.Show("Name cannot be empty.");
                return;
            }

            employees.Add(new Employee(nextId++, name, job, dept));

            RefreshGrid();
            ClearInputs();
        }

```

Figure 3 : Mainform.cs



```csharp
        private void btnEdit_Click(object sender, EventArgs e)
        {
            if (gridEmployees.SelectedRows.Count == 0) return;

            int id = (int)gridEmployees.SelectedRows[0].Cells[0].Value;
            var emp = employees.FirstOrDefault(e => e.Id == id);
            if (emp != null)
            {
                emp.Name = txtName.Text.Trim();
                emp.JobTitle = txtJob.Text.Trim();
                emp.Department = txtDept.Text.Trim();
                RefreshGrid();
            }
        }

        1 reference
        private void btnDelete_Click(object sender, EventArgs e)
        {
            if (gridEmployees.SelectedRows.Count == 0) return;

            int id = (int)gridEmployees.SelectedRows[0].Cells[0].Value;
            employees.RemoveAll(e => e.Id == id);
            RefreshGrid();
            ClearInputs();
        }

        1 reference
        private void gridEmployees_SelectionChanged(object sender, EventArgs e)
        {
            if (gridEmployees.SelectedRows.Count == 0) return;

            int id = (int)gridEmployees.SelectedRows[0].Cells[0].Value;
            var emp = employees.FirstOrDefault(e => e.Id == id);
            if (emp != null)
            {
                txtName.Text = emp.Name;
                txtJob.Text = emp.JobTitle;
                txtDept.Text = emp.Department;
            }
        }
```

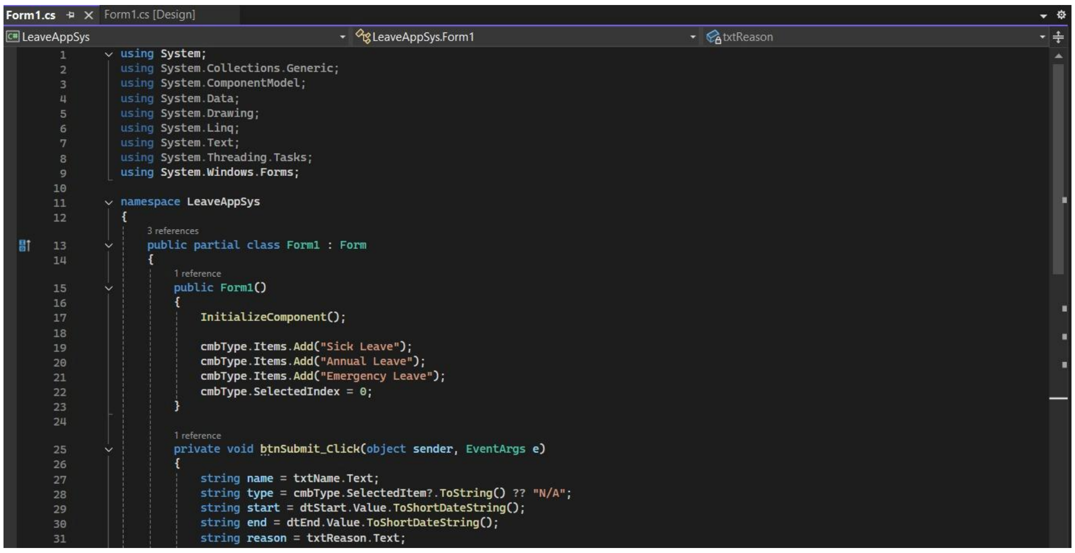
Figure 4: Mainform.cs(cont')



```csharp
        private void RefreshGrid()
        {
            gridEmployees.DataSource = null;
            gridEmployees.DataSource = employees.Select(e => new
            {
                e.Id,
                e.Name,
                e.JobTitle,
                e.Department
            }).ToList();
        }

        2 references
        private void ClearInputs()
        {
            txtName.Clear();
            txtJob.Clear();
            txtDept.Clear();
        }

        1 reference
        private void txtDept_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

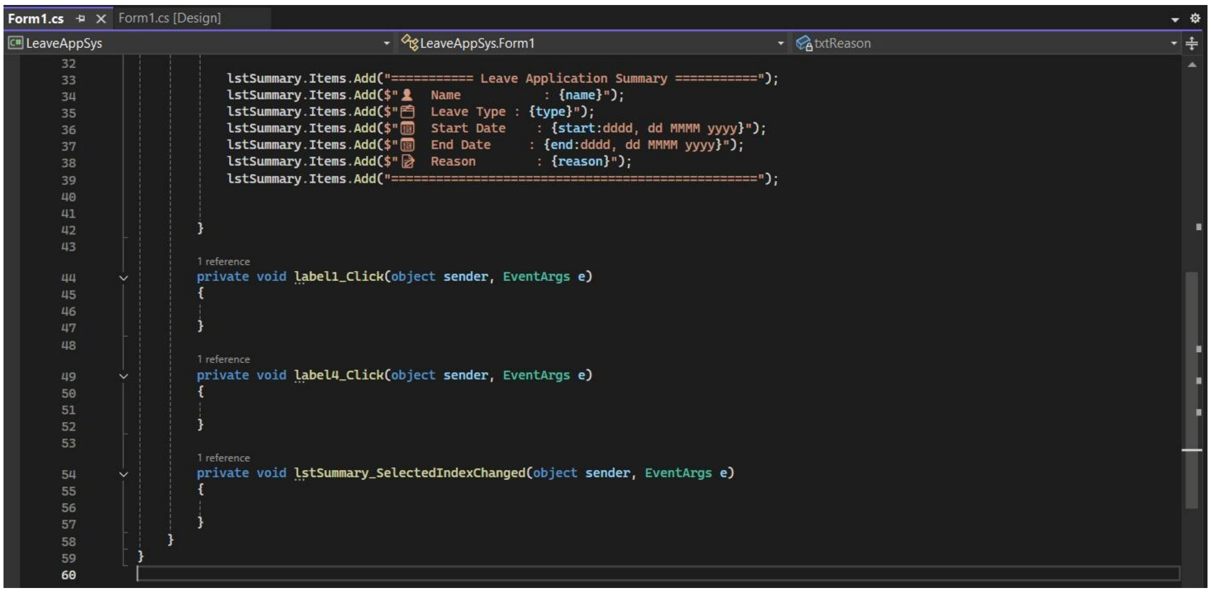
Figure 5: Mainform.cs(cont')

Leave Application



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LeaveAppSys
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();

            cmbType.Items.Add("Sick Leave");
            cmbType.Items.Add("Annual Leave");
            cmbType.Items.Add("Emergency Leave");
            cmbType.SelectedIndex = 0;
        }

        1 reference
        private void btnSubmit_Click(object sender, EventArgs e)
        {
            string name = txtName.Text;
            string type = cmbType.SelectedItem?.ToString() ?? "N/A";
            string start = dtStart.Value.ToShortDateString();
            string end = dtEnd.Value.ToShortDateString();
            string reason = txtReason.Text;
```

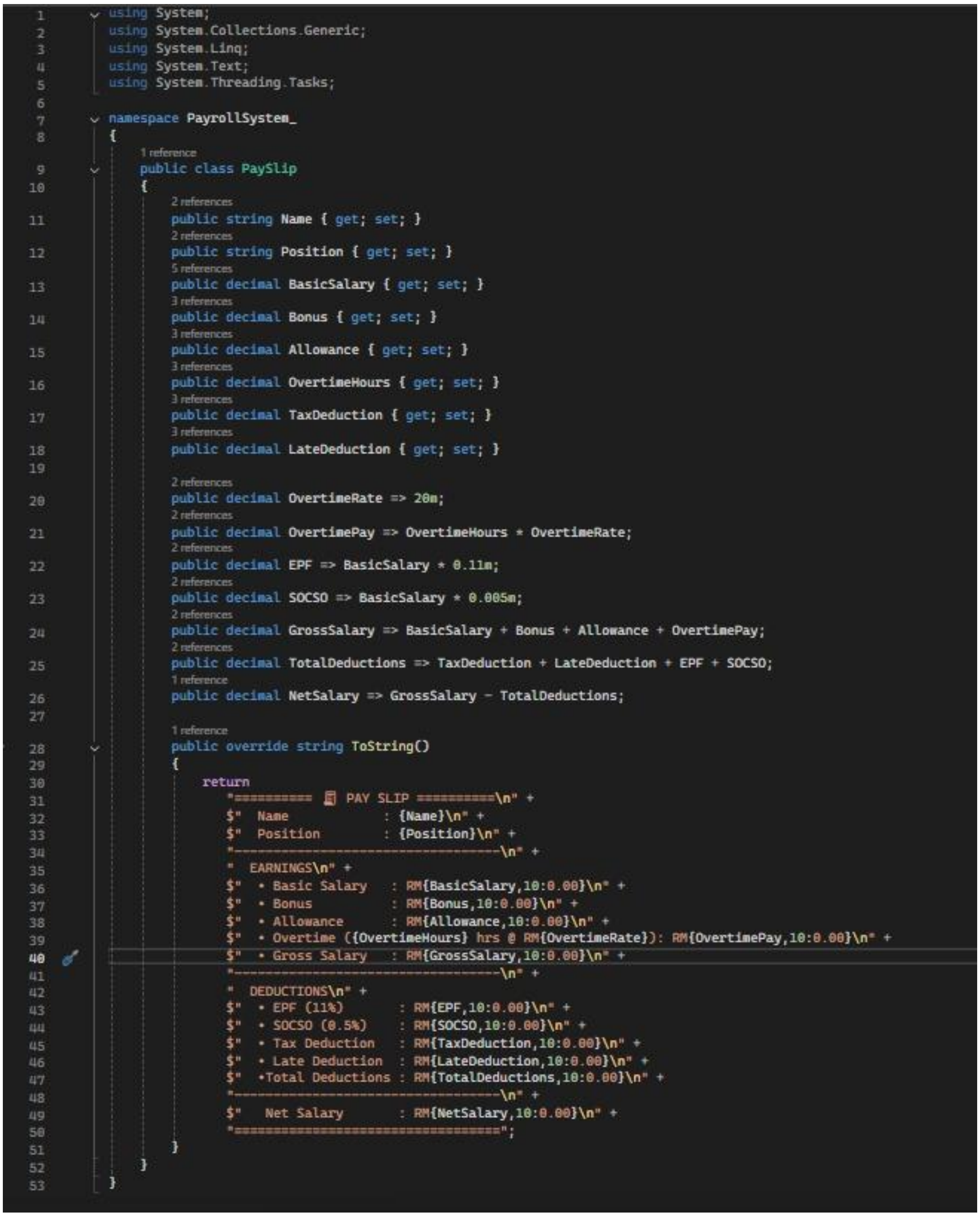
Figure 6: LeaveAppSys.cs



```

            lstSummary.Items.Add("========== Leave Application Summary ==========");
            lstSummary.Items.Add($"👤  Name          : {name}");
            lstSummary.Items.Add($"🗂️  Leave Type : {type}");
            lstSummary.Items.Add($"📅  Start Date    : {start:dddd, dd MMMM yyyy}");
            lstSummary.Items.Add($"📅  End Date     : {end:dddd, dd MMMM yyyy}");
            lstSummary.Items.Add($"📝  Reason        : {reason}");
            lstSummary.Items.Add("================================================");


        }

        1 reference
        private void label1_Click(object sender, EventArgs e)
        {

        }

        1 reference
        private void label4_Click(object sender, EventArgs e)
        {

        }

        1 reference
        private void lstSummary_SelectedIndexChanged(object sender, EventArgs e)
        {

        }
    }
}
```

Figure 7: LeaveAppSys.cs (cont ')

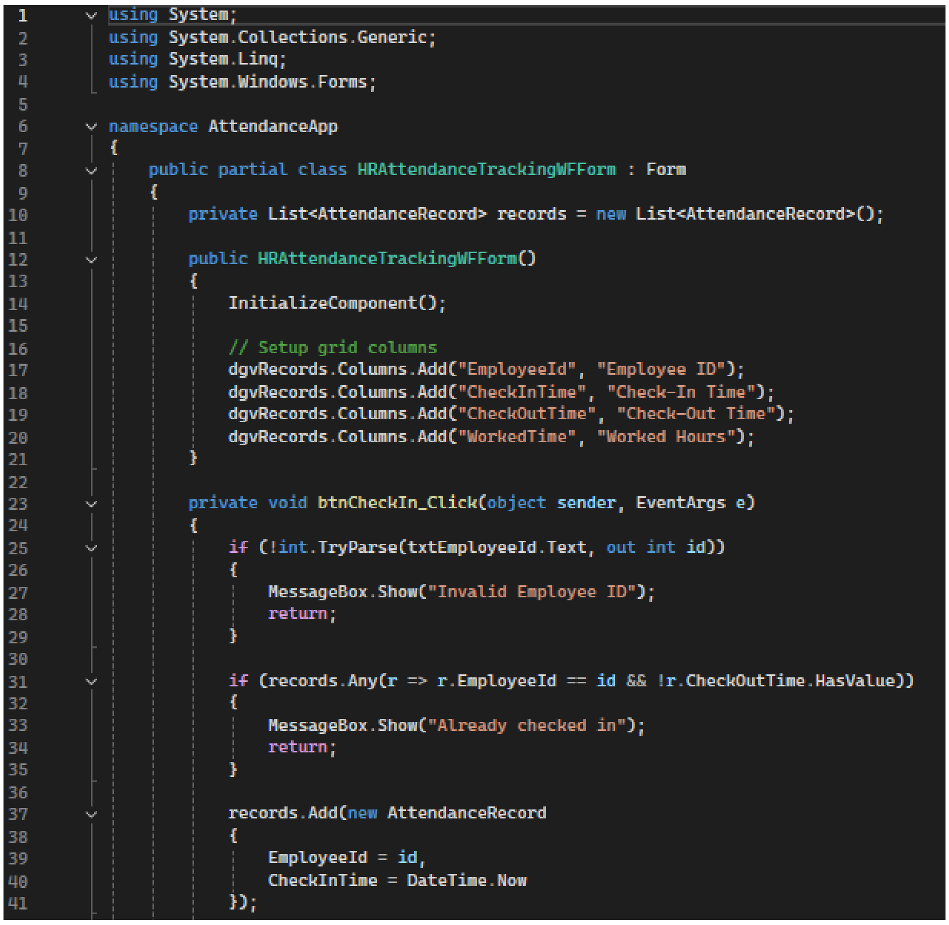
Payroll Module



```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PayrollSystem_
{
    1 reference
    public class PaySlip
    {
        2 references
        public string Name { get; set; }
        2 references
        public string Position { get; set; }
        5 references
        public decimal BasicSalary { get; set; }
        3 references
        public decimal Bonus { get; set; }
        3 references
        public decimal Allowance { get; set; }
        3 references
        public decimal OvertimeHours { get; set; }
        3 references
        public decimal TaxDeduction { get; set; }
        3 references
        public decimal LateDeduction { get; set; }

        2 references
        public decimal OvertimeRate => 20m;
        2 references
        public decimal OvertimePay => OvertimeHours * OvertimeRate;
        2 references
        public decimal EPF => BasicSalary * 0.11m;
        2 references
        public decimal SOCSO => BasicSalary * 0.005m;
        2 references
        public decimal GrossSalary => BasicSalary + Bonus + Allowance + OvertimePay;
        2 references
        public decimal TotalDeductions => TaxDeduction + LateDeduction + EPF + SOCSO;
        1 reference
        public decimal NetSalary => GrossSalary - TotalDeductions;

        1 reference
        public override string ToString()
        {
            return
               "========== 🧾 PAY SLIP ==========\n" +
               $"  Name            : {Name}\n" +
               $"  Position        : {Position}\n" +
               "----------------------------------\n" +
               "  EARNINGS\n" +
               $"  • Basic Salary   : RM{BasicSalary,10:0.00}\n" +
               $"  • Bonus          : RM{Bonus,10:0.00}\n" +
               $"  • Allowance      : RM{Allowance,10:0.00}\n" +
               $"  • Overtime ({OvertimeHours} hrs @ RM{OvertimeRate}): RM{OvertimePay,10:0.00}\n" +
               $"  • Gross Salary   : RM{GrossSalary,10:0.00}\n" +
               "----------------------------------\n" +
               "  DEDUCTIONS\n" +
               $"  • EPF (11%)       : RM{EPF,10:0.00}\n" +
               $"  • SOCSO (0.5%)    : RM{SOCSO,10:0.00}\n" +
               $"  • Tax Deduction   : RM{TaxDeduction,10:0.00}\n" +
               $"  • Late Deduction  : RM{LateDeduction,10:0.00}\n" +
               $"  •Total Deductions : RM{TotalDeductions,10:0.00}\n" +
               "----------------------------------\n" +
               $"   Net Salary       : RM{NetSalary,10:0.00}\n" +
               "==================================";
        }
    }
}
```

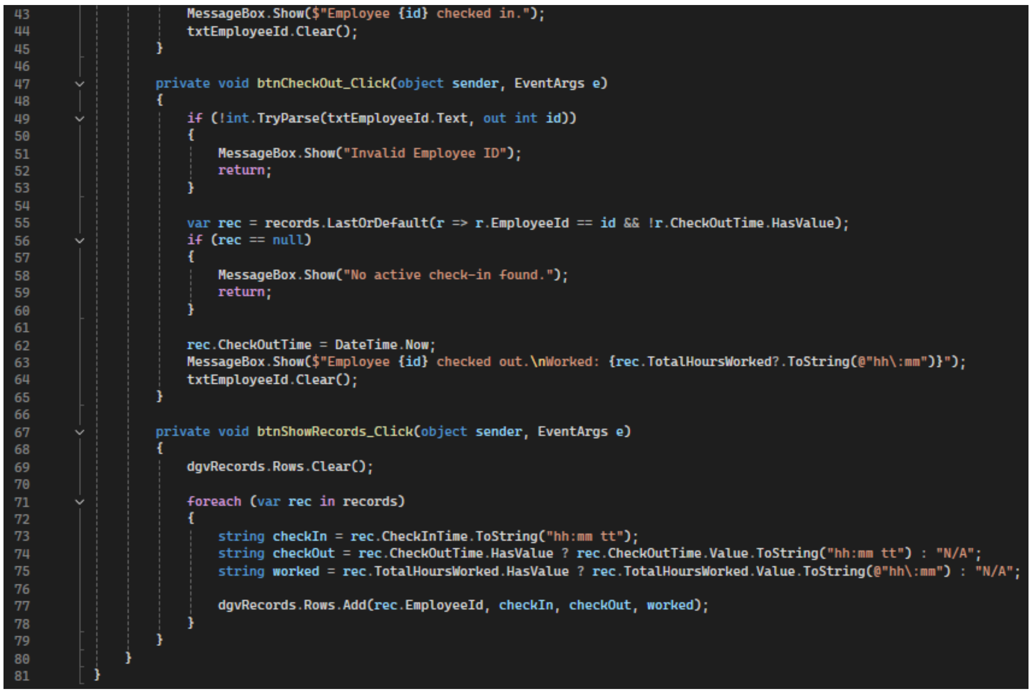
Figure 8:Payslip.cs

Attendance tracking



```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace AttendanceApp
{
    public partial class HRAttendanceTrackingWFForm : Form
    {
        private List<AttendanceRecord> records = new List<AttendanceRecord>();

        public HRAttendanceTrackingWFForm()
        {
            InitializeComponent();

            // Setup grid columns
            dgvRecords.Columns.Add("EmployeeId", "Employee ID");
            dgvRecords.Columns.Add("CheckInTime", "Check-In Time");
            dgvRecords.Columns.Add("CheckOutTime", "Check-Out Time");
            dgvRecords.Columns.Add("WorkedTime", "Worked Hours");
        }

        private void btnCheckIn_Click(object sender, EventArgs e)
        {
            if (!int.TryParse(txtEmployeeId.Text, out int id))
            {
                MessageBox.Show("Invalid Employee ID");
                return;
            }

            if (records.Any(r => r.EmployeeId == id && !r.CheckOutTime.HasValue))
            {
                MessageBox.Show("Already checked in");
                return;
            }

            records.Add(new AttendanceRecord
            {
                EmployeeId = id,
                CheckInTime = DateTime.Now
            });
```

Figure 9: HRAttendanceTracking.cs



```csharp
            MessageBox.Show($"Employee {id} checked in.");
            txtEmployeeId.Clear();
        }

        private void btnCheckOut_Click(object sender, EventArgs e)
        {
            if (!int.TryParse(txtEmployeeId.Text, out int id))
            {
                MessageBox.Show("Invalid Employee ID");
                return;
            }

            var rec = records.LastOrDefault(r => r.EmployeeId == id && !r.CheckOutTime.HasValue);
            if (rec == null)
            {
                MessageBox.Show("No active check-in found.");
                return;
            }

            rec.CheckOutTime = DateTime.Now;
            MessageBox.Show($"Employee {id} checked out.\nWorked: {rec.TotalHoursWorked?.ToString(@"hh\:mm")}");
            txtEmployeeId.Clear();
        }

        private void btnShowRecords_Click(object sender, EventArgs e)
        {
            dgvRecords.Rows.Clear();

            foreach (var rec in records)
            {
                string checkIn = rec.CheckInTime.ToString("hh:mm tt");
                string checkOut = rec.CheckOutTime.HasValue ? rec.CheckOutTime.Value.ToString("hh:mm tt") : "N/A";
                string worked = rec.TotalHoursWorked.HasValue ? rec.TotalHoursWorked.Value.ToString(@"hh\:mm") : "N/A";

                dgvRecords.Rows.Add(rec.EmployeeId, checkIn, checkOut, worked);
            }
        }
    }
}
```

Figure 10: HRAttendanceTracking.cs (cont')

Admin + HR Login

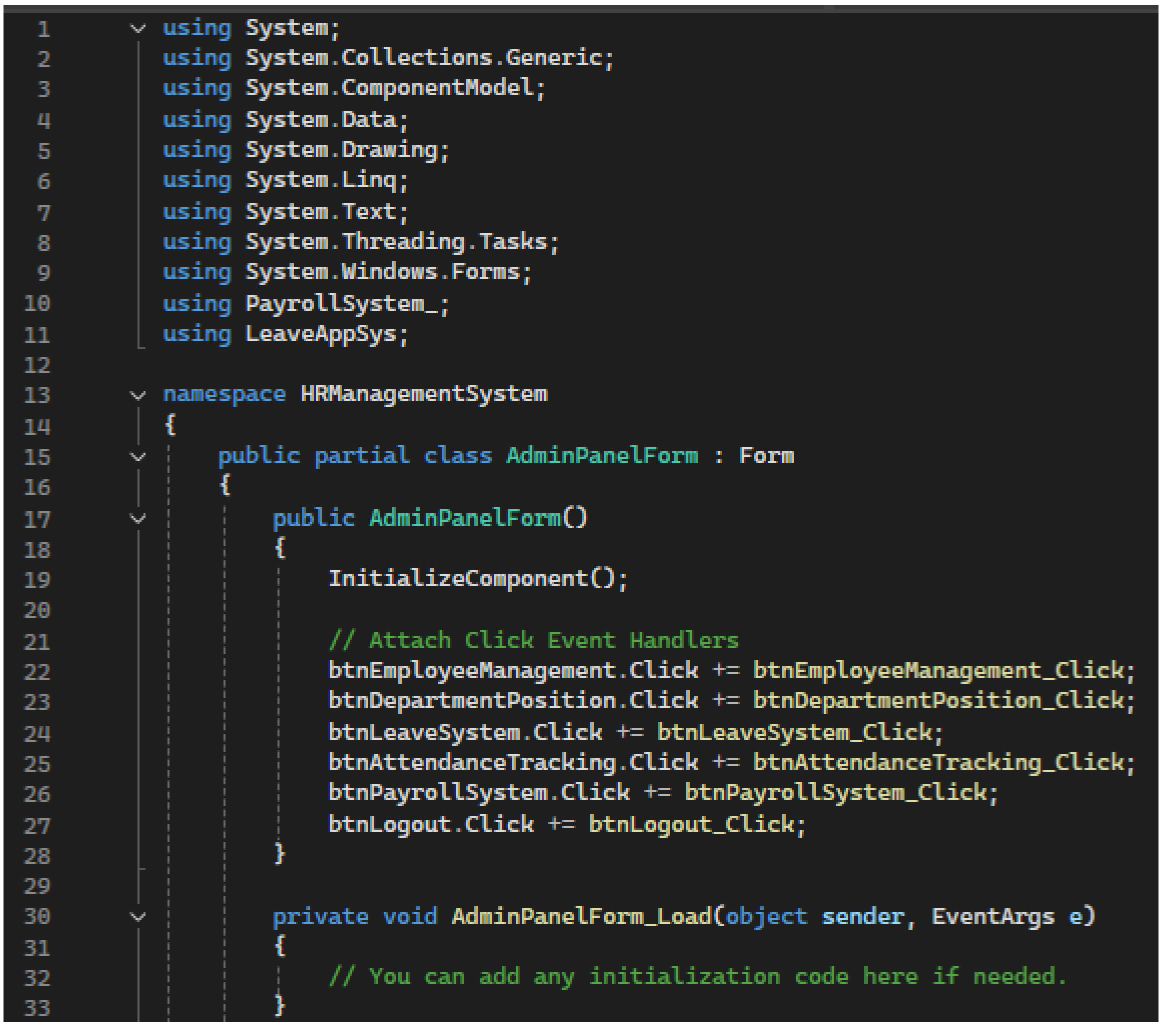
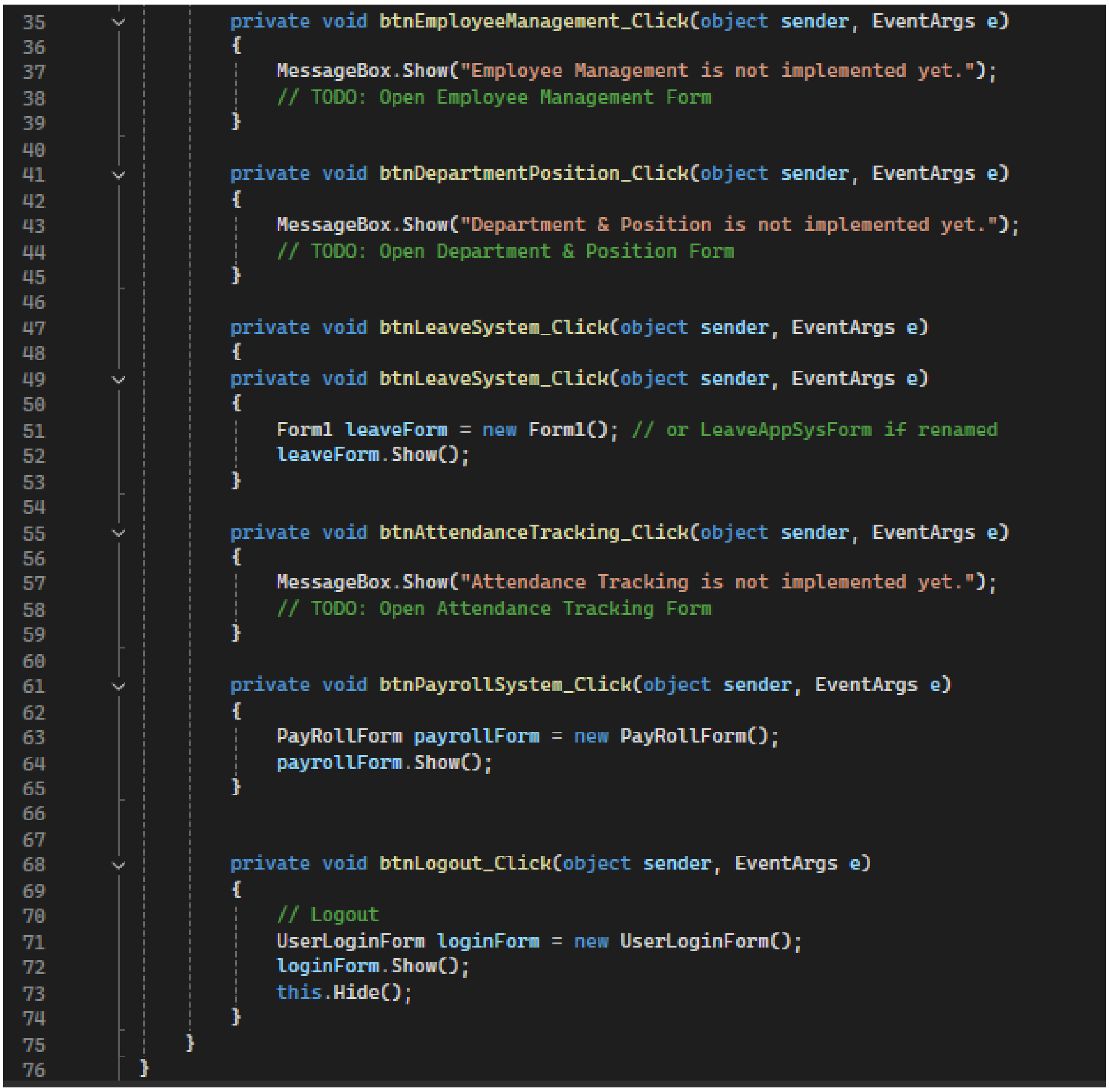

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using PayrollSystem_;
using LeaveAppSys;

namespace HRManagementSystem
{
    public partial class AdminPanelForm : Form
    {
        public AdminPanelForm()
        {
            InitializeComponent();

            // Attach Click Event Handlers
            btnEmployeeManagement.Click += btnEmployeeManagement_Click;
            btnDepartmentPosition.Click += btnDepartmentPosition_Click;
            btnLeaveSystem.Click += btnLeaveSystem_Click;
            btnAttendanceTracking.Click += btnAttendanceTracking_Click;
            btnPayrollSystem.Click += btnPayrollSystem_Click;
            btnLogout.Click += btnLogout_Click;
        }

        private void AdminPanelForm_Load(object sender, EventArgs e)
        {
            // You can add any initialization code here if needed.
        }
```

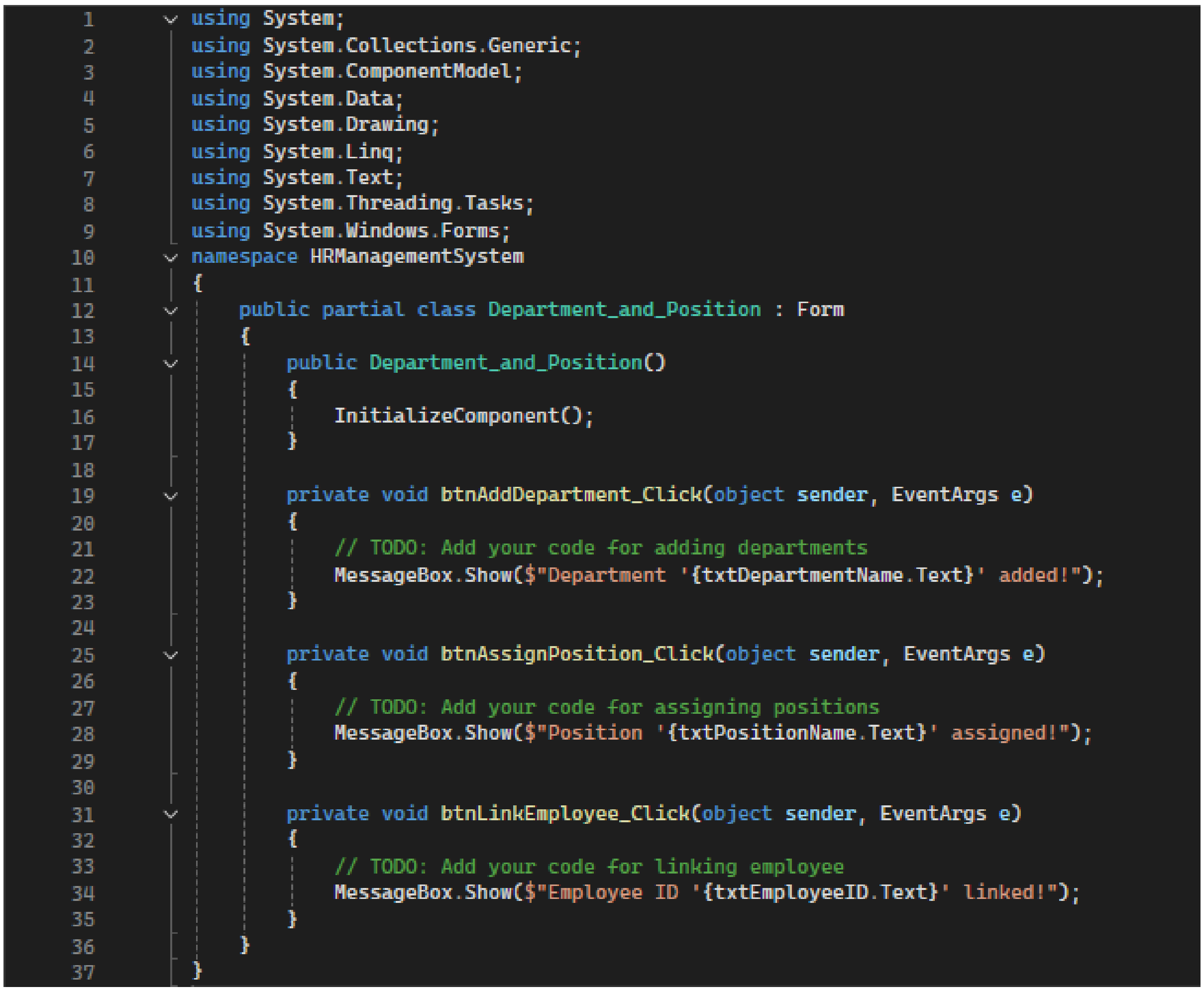
Figure 11: AdminPanelForm.cs



```
        private void btnEmployeeManagement_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Employee Management is not implemented yet.");
            // TODO: Open Employee Management Form
        }

        private void btnDepartmentPosition_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Department & Position is not implemented yet.");
            // TODO: Open Department & Position Form
        }

        private void btnLeaveSystem_Click(object sender, EventArgs e)
        {
        private void btnLeaveSystem_Click(object sender, EventArgs e)
        {
            Form1 leaveForm = new Form1(); // or LeaveAppSysForm if renamed
            leaveForm.Show();
        }

        private void btnAttendanceTracking_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Attendance Tracking is not implemented yet.");
            // TODO: Open Attendance Tracking Form
        }

        private void btnPayrollSystem_Click(object sender, EventArgs e)
        {
            PayRollForm payrollForm = new PayRollForm();
            payrollForm.Show();
        }


        private void btnLogout_Click(object sender, EventArgs e)
        {
            // Logout
            UserLoginForm loginForm = new UserLoginForm();
            loginForm.Show();
            this.Hide();
        }
    }
}
```

Figure 12: AdminPanelForm.cs (cont')

Department & Position

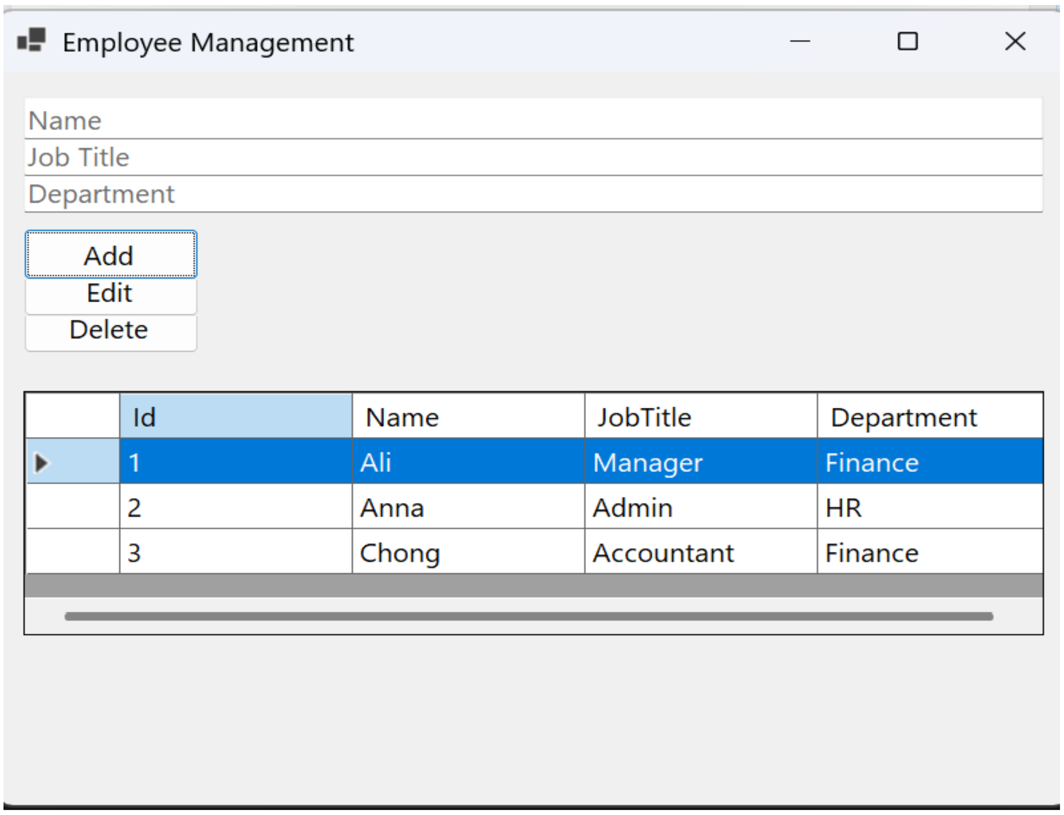


```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace HRManagementSystem
{
    public partial class Department_and_Position : Form
    {
        public Department_and_Position()
        {
            InitializeComponent();
        }

        private void btnAddDepartment_Click(object sender, EventArgs e)
        {
            // TODO: Add your code for adding departments
            MessageBox.Show($"Department '{txtDepartmentName.Text}' added!");
        }

        private void btnAssignPosition_Click(object sender, EventArgs e)
        {
            // TODO: Add your code for assigning positions
            MessageBox.Show($"Position '{txtPositionName.Text}' assigned!");
        }

        private void btnLinkEmployee_Click(object sender, EventArgs e)
        {
            // TODO: Add your code for linking employee
            MessageBox.Show($"Employee ID '{txtEmployeeID.Text}' linked!");
        }
    }
}
```

Figure 13:Department_and_Position.cs

4.2 OUTPUT

Employee Management


Employee Management

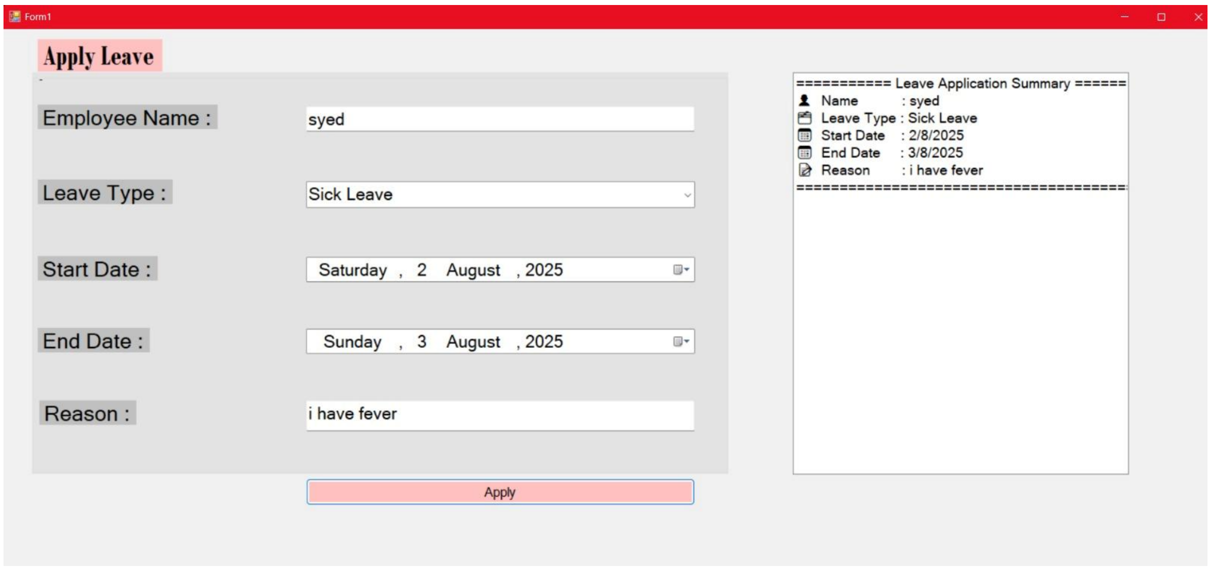
Name
Job Title
Department

Add
Edit
Delete

	Id	Name	JobTitle	Department
▶	1	Ali	Manager	Finance
	2	Anna	Admin	HR
	3	Chong	Accountant	Finance

Figure 14: Employee Management Output

Leave Application


Form1

Apply Leave

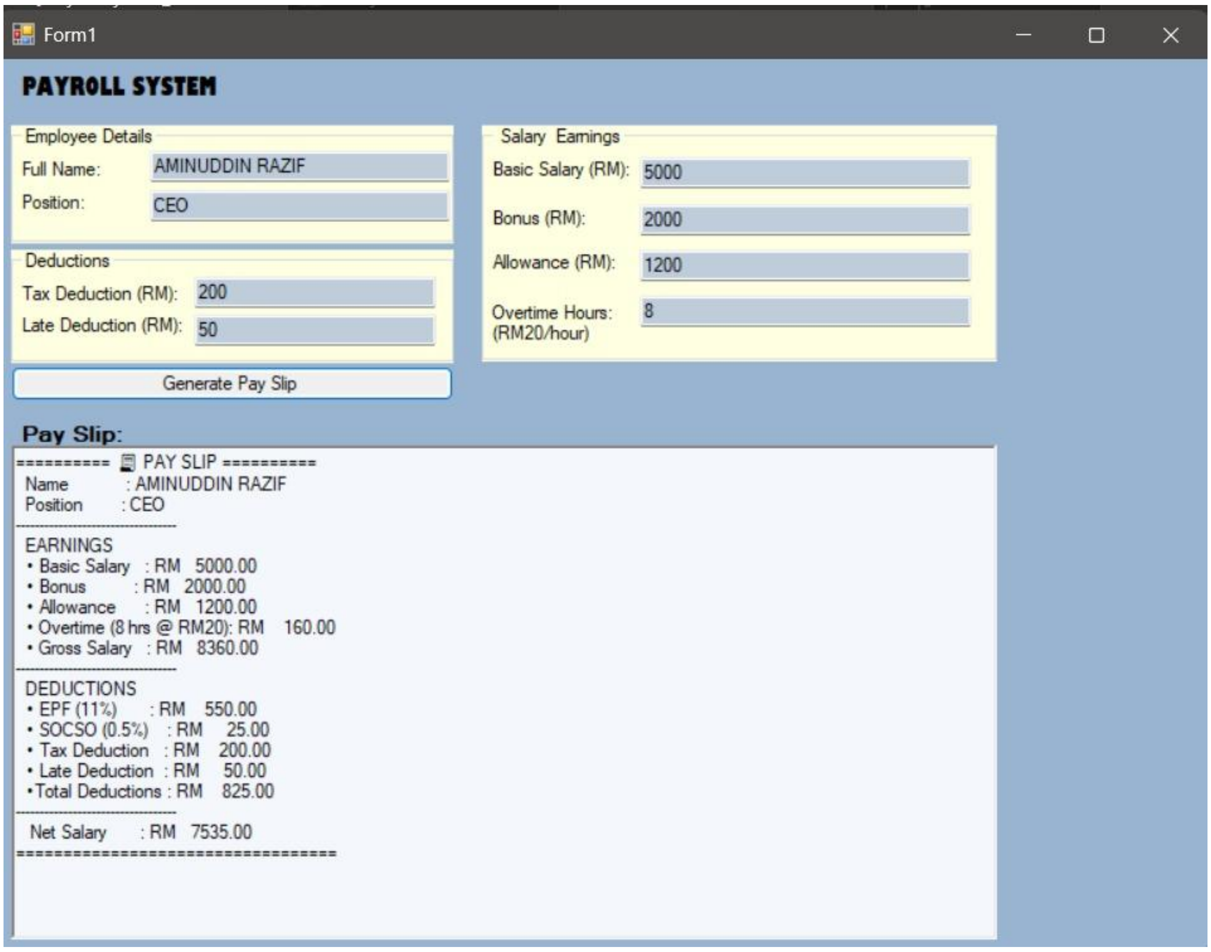
Employee Name :	syed
Leave Type :	Sick Leave
Start Date :	Saturday , 2 August , 2025
End Date :	Sunday , 3 August , 2025
Reason :	i have fever

Apply

```
=========== Leave Application Summary ======
Name       : syed
Leave Type : Sick Leave
Start Date : 2/8/2025
End Date   : 3/8/2025
Reason     : i have fever
=========================================
```

Figure 15: Leave Application Output

Payroll Module



Full Name: AMINUDDIN RAZIF
Position: CEO
Tax Deduction (RM): 200
Late Deduction (RM): 50
Basic Salary (RM): 5000
Bonus (RM): 2000
Allowance (RM): 1200
Overtime Hours: (RM20/hour) 8

```
========== PAY SLIP ==========
 Name         : AMINUDDIN RAZIF
 Position     : CEO
---------------------
 EARNINGS
 • Basic Salary   : RM   5000.00
 • Bonus          : RM   2000.00
 • Allowance      : RM   1200.00
 • Overtime (8 hrs @ RM20): RM    160.00
 • Gross Salary   : RM   8360.00
---------------------
 DEDUCTIONS
 • EPF (11%)       : RM    550.00
 • SOCSO (0.5%)    : RM     25.00
 • Tax Deduction   : RM    200.00
 • Late Deduction  : RM     50.00
 •Total Deductions : RM    825.00
---------------------
  Net Salary      : RM   7535.00
==============================
```

Figure 16: Payroll Module Output

Attendance tracking

HR Login

Department & Position