DAY 5 Assignment UML Diagrams

BY

Jonnagiri siva naga prasanna

28 Jan 2022

|  |
| --- |
| **AMAZON** |

|  |
| --- |
| **Employee Class** |
| Code |
| class Employee  {  private int Eid;  private string Name;  private string Address;  private double salary;  private string emailid;  }  public void AddEmployeeDetails()  {  //Todo  }  public void EditEmployeeDetails()  {  //Todo  }  public DisplayEmployeeDetails()  {  //Todo  }  public SelectEmployeeDetails()  {  //Todo  }  public DeleteEmployeeDetails()  {  //Todo  } |
| UML Diagram |
| Employee  - Eid:int  - Address:string  - Salary:int  - EmailID:string  +AddEmployeeDetails() :void()  +EditEmployeeDetails() : void()  +DisplayEmployeeDetails() : void()  +SelectEmployeeDetails() :void()  +DeleteEmployeeDetails():void() |

|  |
| --- |
| **Product Class** |
| Code |
| class Product  {  private int Pid;  private string PName;  private string Price;  private string Model;  private string Brand;  }  public void SelectProduct()  {  //Todo  }  public void AddProduct()  {  //Todo  }  public EditProduct()  {  //Todo  }  public DisplayProduct()  {  //Todo  }  public DeleteProduct()  {  //Todo |
| **UML Diagram** |
| Product   * Pid:int * Price:int * Model:string * Brand:string   + SelectProduct(): void()  +AddProduct () : void()  +EditProduct() : void()  +DisplayProduct () :void()  +DeleteProduct – void()  -AddProduct()  -DeleteProduct()  -Display product()  -Selectproduct()  -Editproduct() |

|  |
| --- |
| **Order class** |
| Code |
| class Order  {  private int Oid;  private string Name;  private short PhoneNumber;  private string Pincode;  private string Landmark;  }  public void AddOrder()  {  //Todo  }  public void EditOrder()  {  //Todo  }  public SelectOrder()  {  //Todo  }  public DisplayOrder()  {  //Todo  }  public DeleteOrder()  {  //Todo  } |
| **UML Diagram** |
| Order   * Oid:int * Name:string:int * Pincode:int * Landmark:string   +AddOrder () :void()  +EditOrder() : void()  +SelectOrder() : void()  +DisplayOrder(): void()  +DeleteOrder(): void() |

|  |
| --- |
| **Customer** |
| Code |
| class Customer  {  private string CustName;  private int Custid;  private short PhoneNumber;  private string Address;  private string Credit Rating;  }  public void AddCustomer()  {  //Todo  }  public void EditCustomer()  {  //Todo  }  public SelectCustomer()  {  //Todo  }  public DisplayCustomer()  {  //Todo  }  public DeleteCustomer()  {  //Todo  } |
| **UML Diagram** |
| Customer   * CustName:string * Cust ID:int * Phone Number:int * Address:string * Credit Rating:int   +AddCustomer () : void()  +EditCustomer() : void()  +SelectCustomer() : void()  +DisplayCustomer():void()  +DeleteCustomer():void() |

|  |
| --- |
| **Transactions** |
| Code |
| class Transactions  {  private string TransID;  private string HolderName;  private string Cardtype;  private int Accountno.;  private string EmailID;  }  public void AddTransaction()  {  //Todo  }  public void EditTransaction()  {  //Todo  }  Public void SearchTransaction()  {  //Todo  }  public DisplayTransaction()  {  //Todo  }  public DeleteTransaction()  {  //Todo  } |
| **UML Diagrams** |
| Transaction   * TransID:int * HolderName:string * CardTpye:int:string * EmailID:string   + AddTransaction(): void()  +EditTransaction() : void()  +SearchTransaction() :void()  +DisplayTransaction() :void()  +DeleteTransaction(): void() |

|  |
| --- |
| **Police Station** |

|  |
| --- |
| **Police** |
| Code |
| class Police  {  private string Name;  private int Id;  private string Designation;  private short PhoneNumber;  private string Department;  private string EmailID;  }  public void AddPolicedata()  {  //Todo  }  public void EditPolicedata()  {  //Todo  }  public void DisplayPolicedata()  {  //Todo  }  public void DeletePolicedata()    {  //Todo  }  Public void UpdatePoliceData()  {  //Todo  } |
| **UML Diagram** |
| Police   * Name:string * ID:int * Designation:string * PhoneNumber:int * Department:string * EmailID:string   +AddPolicedata() :void()  +EditPolicedata() : void()  +DisplayPolicedata() : void()  +DeletePolicedata(): void()  +UpdatePolicedata():void() |

|  |
| --- |
| **Criminals** |
| Code |
| class Criminals  {  private string Name;  private string Id;  private string Crime\_Type;  private short PhoneNumber;  private string Address;  }  public void AddCriminals()  {  //Todo  }  public void EditCriminals()  {  //Todo  }  public void DisplayCriminals()  {  //Todo  }  public void DeleteCriminals()  {  //Todo  }  Public void SearchCriminals()  {  //Todo  } |
| **UML Diagrams** |
| Criminals   * Name:string * ID:int * CrimeTpe:string * PhoneNUmber:int * Address:string   +AddCriminals() :void()  +EditCriminals() : void()  +DisplayCriminals() : void()  +DeleteCriminals():void()  +SearchCriminals():void() |

|  |
| --- |
| **Prisoner** |
| Code |
| class Prisoner  {  private string PName;  private int P\_id;  private string EmailID;  private string PhoneNumber;  private string Address;    }  public void AddPrisoner()  {  //Todo  }  public void EditPrisoner()  {  //Todo  }  public SearchPrisoner()  {  //Todo  }  public DisplayPrisoner()  {  //Todo  }  public DeletePrisoner()  {  //Todo  } |
| **UML Diagram** |
| Prisoners   * PName:string * P\_ID:int * EmailID:string:int * Address:string   +AddPrisoner() :void()  +EditPrisoner() : void()  +SearchPrisoner() : void()  +DisplayPrisoner():void()  +DeletePrisoner():void() |

|  |
| --- |
| **Fir** |
| Code |
| class Fir  {  private string Name;  private int id;  private string Type;  private short PhoneNumber;  private string Description;    }  public void AddFIR()  {  //Todo  }  public void EditFIR()  {  //Todo  }  public SearchFIR()  {  //Todo  }  public DisplayFIR()  {  //Todo  }  public void DeleteFIR()  {  //Todo  } |
| **UML Diagram** |
| Fir   * Name:string * ID:int * Type:string * PhoneNumber:int * Address:string   +AddFir() :void()  +EditFir() : void()  +SearchFir() : void()  +DisplayFir():void()  +DeleteFir():void() |

|  |
| --- |
| **Complaints** |
| Code |
| class Complaints {  private string Name;  private string Issue;  private string PhoneNumber;  private string Address;  private string Writter;    }  public void AddComplaints()  {  //Todo  }  public void EditComplaints()  {  //Todo  }  Public void SearchComplaints()  {  //Todo  }  Public void DisplayComplaints()  {  //Todo  }  public void DeleteComplaints()  {  //Todo  } |
| **UML Diagram** |
| Complaints   * Name:string * Issue:string * PhoneNumber:int * Address:string * Writter:string   +AddComplaint() :void()  +EditComplaint() : void()  +SearchComplaint() : void()  +DisplayComplaint():void()  +DeleteComplaint():void() |

|  |
| --- |
| **APOLLO HOSPITAL** |

|  |
| --- |
| **Doctors** |
| Code |
| class Doctor  {  private string Name;  private string Id;  private string Designation;  private short PhoneNumber;  private string EmailID;  }  public void AddDoctorsData()  {  //Todo  }  public void ModifyDoctorsData()  {  //Todo  }  public void DisplayDoctorsData()  {  //Todo  }  public void DeleteDoctorsData()  {  //Todo  }  public void UpdateDoctorsData()  {  //Todo  } |
| **UML Diagram** |
| **Doctors**   * Name:string * ID:int * Designation:string * PhoneNumber:int * EmailID:string   +AddDoctorsData() :void()  +ModifyDoctorsData() : void()  +DisplayPoliceData() : void()  +DeletePoliceData(): void()  +UpdatePoliceData():void() |

|  |
| --- |
| **Nurse** |
| Code |
| class Nurse  {  private string Name;  private string Id;  private string Category;  private short PhoneNumber;  private string EmailID;  }  public void AddNurseData()  {  //Todo  }  public void EditNurseData()  {  //Todo  }  public void SearchPatients()  {  //Todo  }  public void DisplayNurseData()  {  //Todo  }  public void DeleteNurseData()  {  //Todo  } |
| **UML Diagram** |
| **Nurse**   * Name:string * ID:int * Category:string:int * PhoneNumber:int * EmailID:string   +AddNurseData() :void()  +EditNurseData() : void()  + SearchPatients(): void()  +DisplayNurseData() : void()  +DeleteNurseData(): void() |

|  |
| --- |
| **Patients** |
| Code |
| class Patient  {  private string Name;  private string Id;  private string Gender;  private short PhoneNumber;  private string CardType;  private string Address;  }  public void AddPatientDetails()  {  //Todo  }  public void EditPatientDetails ()  {  //Todo  }  public void SearchPatientDetails ()  {  //Todo  }  public void DisplayPatientDetails ()  {  //Todo  }  public void DeletePatientsDetails()  {  //Todo  } |
| **UML Diagram** |
| **Patients**   * Name:string * ID:int * Gender:string * PhoneNumber:int * CardType:string * AppointmentDate:int   +AddPatientDetails() :void()  +EditPatientDetails() : void()  + SearchPatientDetails(): void()  +DisplayPatientDetails() : void()  +DeletePatientDetails(): void()  +UpdateNurseData():void() |

|  |
| --- |
| **Staff** |
| Code |
| class Staff  {  private string Name;  private string Id;  private int Joining\_Date;  private string Gender;  private string Blood\_Group;  private short PhoneNumber;    }  public void AddStaffData()  {  //Todo  }  public void EditStaffData()  {  //Todo  }  public void SearchStaffData()  {  //Todo  }  public void DeleteStaffData()  {  //Todo  }  public void DisplayStaffData()  {  //Todo  } |
| **UML Diagram** |
| **Staff**   * Name:string * ID:int * Joining\_Date :int * Gender:string * Blood\_Group:string * PhoneNumber:int   +AddStaffDetails() :void()  +EditStaffData() : void()  + SearchStaffData(): void()  +DeleteStaffData() : void()  +DisplayStaffData(): void()  +UpdateNurseData():void() |

|  |
| --- |
| **Medicine** |
| Code |
| class Hospital  {  private string Tab\_Name;  private string Description;  private int Manufacturing\_Date;  private int Batchno.;  private int Expiry\_Date;    }  public void AddMedicinel()  {  //Todo  }  public void EditMedicine()  {  //Todo  }  public SearchMedidcine()  {  //Todo  }  public DisplayMedicine()  {  //Todo  }  public void DeleteMedicine()  {  //Todo  } |
| **UML Diagram** |
| **Medicine**   * TabName:string * Description:string * Manufacturing\_Date:int * Batchno.:int * Expiry\_Date:int   +AddMedicine() :void()  +EditMedicine() : void()  +SearchMedicine(): void()  +DisplayMedicine() : void()  +DeleteMedicine(): void()  : |