**Pecunia Finance Limited – Case study**

Team E – Classes Document

* Asmita
* Tarunsree
* Akash
* Siddharth
* Shobhit

Class 1: Employee

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private int employeeID | Primary key (unique) value of employee. |
| 2 | private string employeeName | Name of the employee |
| 3 | private string employeeCode | Unique code of the employee |
| 4 | Private string emailId | Email Id of the employee |
| 5 | Private string employeePassword | Password of the employee |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | public int EmployeeID | Unique; not null; System-generated (starts with 1, increment by 1) |
| 2 | public string EmployeeName | Name of the employee.  Alpha-numeric; not null.  Should be less than 40 characters. |
| 3 | public string EmployeeCode | Unique; Not null.  Should between 4 to 10 characters.  Alpha-numeric. |
| 4 | Public string EmailId | Unique, Not null,  should contain @ |
| 5 | Public string EmployeePassword | Not Null, 8-10 characters. Alpha numeric, Should contain atleast one capital letter and should contain special character |

Class 2: EmployeesService

**Methods:**

|  |  |  |
| --- | --- | --- |
| **#** | **Method** | **Description** |
| 1 | public List<Employee> GetEmployees() | Returns entire list of employees sorted on employee name. |
| 2 | Public void AddEmployee() | Adds a new employee when new employee joins |
| 3 | public Employee GetEmployeeByUserNameAndPassword(string Username, string Password) | Returns list of employees that matches with given username and password. |
| 4 | public Employee GetEmployeeByEmail(string Email) | Return single employee object that matches with given email address.  Return null if not matched. |

Class 3: Customer

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private int customerID | Primary key (unique) value of customer. |
| 2 | private string customerName | Name of the customer |
| 3 | private string phone | Mobile no. of customer |
| 4 | Private string address | Address of customer |
| 5 | Private string emailID | Email ID of customer |
| 6 | Private DateTime dob | DOB of customer in date time format |
| 7 | Private string aadharNo | Aadhar no.of customer |
| 8 | Private string panNo | PAN no. of customer |
| 9 | Private char gender | Gender of customer |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | public int CustomerID | Unique; not null; System-generated (starts with 4, increment by 1) |
| 2 | public string CustomerName | Name of the employee.  not null.  Should be less than 40 characters. |
| 3 | public string Phone | Unique; Not null.  Should be of 10 digits. |
| 4 | Public string Address | Should be less than 200 characters |
| 5 | Public string EmailId | Should Contain @ and end with .com |
| 6 | Public string DOB | Should be in YYYY/MM/DD format |
| 7 | Public string AadharNO. | Should be of 12 digits only |
| 8 | Public string PAN\_No. | Should have first 5 letters then next 4 digits and last one should be letter.so total 10 |
| 9 | Public char gender | Only M/m and F/f should be allowed |

**Class 4: CustomerServiceClass**

|  |  |  |
| --- | --- | --- |
| **#** | **Method** | **Description** |
| 1 | public List<customer> Customer | Returns entire list of existing customer |
| 2. | Public void Add\_Customer(NAME , AGE , PHONE ETC) | Will ADD new customer to the list |
| 3. | Public void Update\_Customer(Customer\_id) | Will update customer by using customer id |
| 4. | Public Delete\_Customer(customer\_id) | Will delete the customer details using customer id |
| 5. | Public void Get\_Customer\_details() | Will display customer details based on customer id or customer name or as required |

Class 5: ACCOUNTS

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | Private string accountno | Primary key (unique) |
| 2 | Private double balance | Current balance of account |
| 3 | private DateTime startdate | Date of Account creation |
| 4 | Private string accountType | Savings or current or Fixed deposit |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | public string Accountno | Unique; not null; System-generated (starts with 1000000000, increment by 1) |
| 2 | public double Balance | It should be greater than minimum balance |
| 3 | Public DateTime StartDate | Not null |
| 4 | Public string AccountType | Not null  Should be Savings or current or Fixed deposit |

Class 6: IAccountsService

**Methods:**

|  |  |  |
| --- | --- | --- |
| **#** | **Method** | **Description** |
| 1 | public abstract void CreateAccount() | Create account and generate new account number |
| 2 | Public abstract void CloseAccount() | Adds a new employee when new employee joins |
| 3 | public abstract void UpdateAccount() | Returns list of employees that matches with given username and password. |
| 4 | Public abstract double Show Balance(accountno) | In case of any transaction update the balance |
| 5 | public abstract List<Account> ListAccountsBytype(string AccountType) | Create a list of accounts by type and return list of objects |

CHILD CLASS- Savings account: IAccountsService

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private static double minimumbalance | Minimum balance of savings account |
| 2 | Private double interestRate | Interest rate of savings account |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | Public static double MinimumBalance | Not negative |
| 2 | public double InterestRate | It should be greater than 0 |

CHILD CLASS-Current account: IAccountsService

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private static double minimumBalance | Minimum balance of current account |
| 2 | Private double interestRate | Interest rate of current account |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | public double MinimumBalance | Greater than 500 |
| 2 | public double InterestRate | It should be greater than 0 |

CHILD CLASS- Fixed deposit account: IAccountsService

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private double minimumBalance | Minimum balance of current account |
| 2 | Private double interestRate | Interest rate of current account |
| 3 | Private double tenure | Tenure of the deposit |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | Public double MinimumBalance | Not negative |
| 2 | Public double InterestRate | It should be greater than 0 and differ with age and gender |
| 3 | Public double Tenure | It should not be negative, should be selected from options given by bank |

**Methods:**

|  |  |  |
| --- | --- | --- |
| **#** | **Method** | **Description** |
| 1 | public void calculate interest(double InteresTRate, double balance, double tenure) | This method will calculate the interest using interest rate , tenure and the balance in account |

Class 7: Loans

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private string loantype | Type of loan home, personal etc |
| 2 | private double loan\_amt | Amount of loan |
| 3 | private double collateral | Collateral submitted on loan applied |
| 4. | Private string loan status | Current loan status (applied , approved) |
| 5. | Private int tenure | Tenure left of loan |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | public string loantype | It can be home personal education or vehicle |
| 2 | public int loan\_amt | Should be greater than 100000 |
| 3 | public double collateral | Collateral should be greater than loan amount |
| 4. | Public string loan status | It can be applied approved rejected pending |
| 5. | Public double tenure | Should be greater than 0 |
| 6. | Public DateTime dateOfLoanApplied | Date of loan Applied in format YYYY/MM/DD |
| 7. | Public List<loan>LoanApplicant | Will contain list of all loan applicant |
| 8. | Public list<Loan>ApprovedLoan | Will contain list of all approved and loans |
| 9. | Public List<Loan>RejectedLoan | Will Contain loans of all Rejected loans |
| 10. | Public List<loan>InProgressLoan | Will return loan details of loan which are in progress |
| 11. | Public List<loan>PendingLoan | Will return list of all pending Loan |

Class 8: LoansService

**Methods:**

|  |  |  |
| --- | --- | --- |
| **#** | **Method** | **Description** |
|  | Public List<Loan>Approved\_loan(string loanStatus) | Gives list of loan approved EMI Tenure , balance left |
| 2. | Public List<Loan>Rejected Loan(string loanStatus) | Gives list of rejected loan and EMI amount or other requirments |
| 3. | Public Void Add\_New\_Loan() | Will take details to apply for new loan ..will be linked to customer class |
| 4. | Public Void Update() | Will Update amount tenure and other details |
| 5. | Public bool Approval(double salary , double collateral , double tenure) | Will decide whether Loan is approved or not |
| 6. | Public double EMI(double salary , double collateral) | Will Calculate EMI for Loan |
| 7. | Public Void GetLoanByCustomerId(string Customer\_Id) | Will return customer and loan details based on customer id |
| 8. | Public list<loan> LoanDetailsByDate(DateTime ) | Will return list of all loan details based on date of loan applied |
| 9. | Public list<loan>InProgressLoan(string loanStatus) | Will return details of all in progress loan |
| 10. | Public list<loan>PendingLoan(string loanStatus) | Will return details of all pending loan |

Class 7: Transaction

**Fields:**

|  |  |  |
| --- | --- | --- |
| **#** | **Field** | **Description** |
| 1 | private string creditAccountNo | Credit account number. |
| 2 | private string debitAccountNo | Debit Account number. Should be in database |
| 3 | private double amount | Amount of money to be transferred. |
| 4 | private string transactionType | Debit/Credit |

**Properties:**

|  |  |  |
| --- | --- | --- |
| **#** | **Property** | **Validations** |
| 1 | public string CreditAccountNo.( ) | Get credit account no. Should be in database |
| 2 | public string DebitAccountNo.() | Get debit account number. Should be in database |
| 3 | public string Amount() | Get Amount of transaction. Should be greater than 0. |
| 4. | Public List<Transaction>GetTransactionByAccountNo | Gives a list of transaction for a particular account |
| 5. | Public List<Transaction>GetTransactionByTransactionType | Gives a list of transaction for a particular transaction type |

Class 8: TransactionsService

**Methods:**

|  |  |  |
| --- | --- | --- |
| **#** | **Method** | **Description** |
| 1. | Public double Credit( string AccountNumber, double amount) | Update the give account balance by crediting the given amount |
| 2. | Public double Debit( string AccountNumber, double amount) | Update the given account number by debit by debiting the given account |
| 3. | Public void Transfer( string creditAccountNo, string debitAccountNo, double amount) | Transfers the given amount from debit account to credit account and updates the current balance of both accounts |
| 3. | Public void GetTransactionBydate( DateTime StartDate, DateTime EndDate) | Gives a list of transaction for a range of dates |
| 4. | Public List<Transaction>GetTransactionByAccountNo | Gives a list of transaction for a particular account |
|  |  |  |