This project is implemented by .Net 4.8 framework and it is a console application.

The Models folder has model classes that are used to hold data and some enum types.

The Services folder has a salary calculation service which breakdowns the cross salary package into superannuation and taxable income and then uses the taxable income to call deduction lookup services to calculate deductions. Once the deductions get calculated, the net income will be able to be calculated. Then according to the pay frequency, it calculates the pay packet amount. In short, the salary calculation service to calculate the salary details data using the cross salary and pay frequency provided by the user.

The Service folder also has a factory deduction service. A factory method pattern is used to return different deduction lookup services according to the deduction type e.g. Medicare levy, budget repair levy, income tax. All deduction services implement the same interface . A factory method design pattern can separate concerns and is extendable. If a new deduction type is required, only need to add a new class to implement the new rules without affecting the existing deduction services. If one of the existing deductions have any change, it won’t affect other deduction services.

The Main method in the Program class receives the cross salary and pay frequency data the user enters and calls the salary calculation service to calculate salary details data. The salary details data is held by an object of SalaryComponentModel class. Then the Main method displays the salary details on console.

Test cases:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated