Name: Sampreet Klair

Student Number: 145031225

Student Email: [sklair2@myseneca.ca](mailto:sklair2@myseneca.ca)

Date: 31st January 2024

1. Create a class diagram to support the above case study and Systems Use Case Specifications
2. Create an object level sequence diagram, detailing the Create Income Transaction systems use case specification
3. Create an object level sequence diagram, detailing the Create Expense Transaction systems use case specification
4. Create an object level sequence diagram, detailing the Query Financial Transaction system use case specification.
5. Analyze your model and outline what the .h files would look like if you generated code from your model

Answer 1:

A screenshot of a diagram

Description automatically generated

A diagram of a server

Description automatically generated

Answer 2:

A diagram of a project

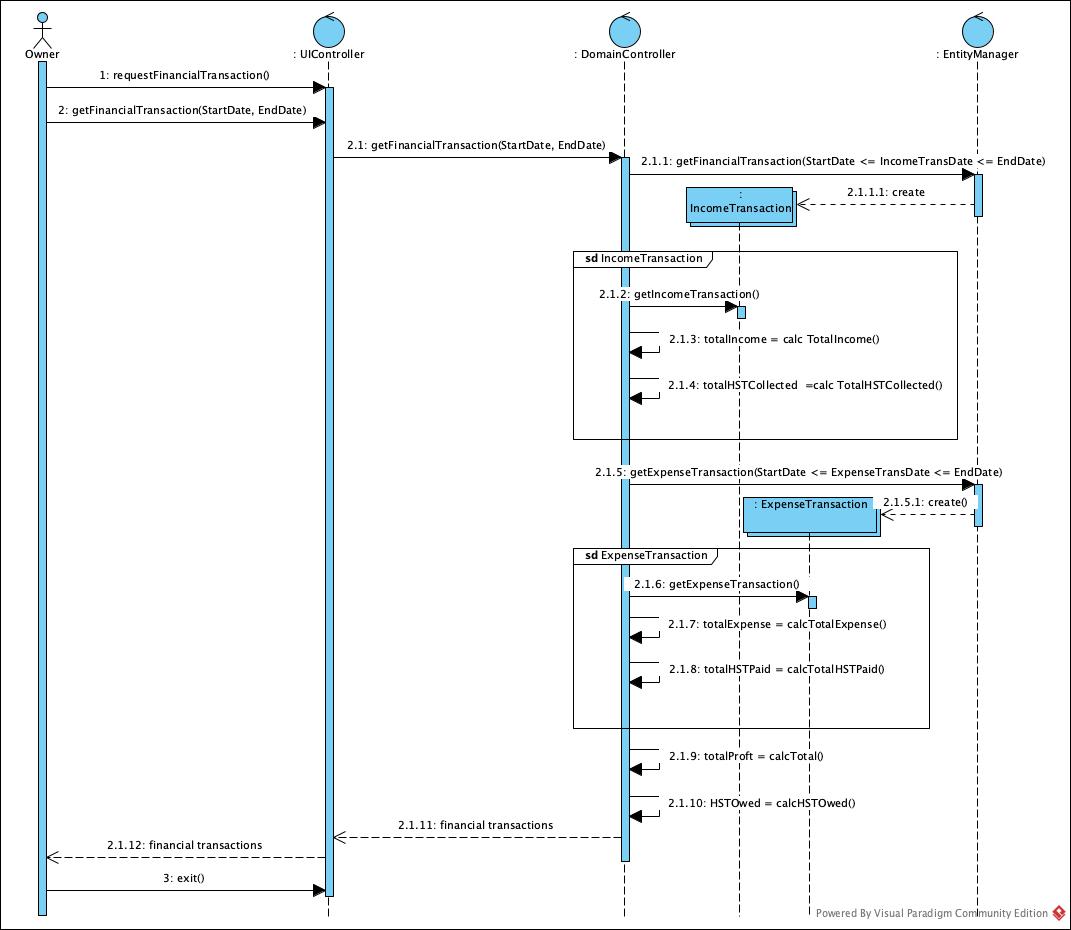
Description automatically generated

Answer 3:

A diagram of a project

Description automatically generated

Answer 4:



Answer 5:

Public class EntityManager(){

getPaymentTypes(){

//retrieves payment types from the data source

}

getFinancialTransaction(){

//retrieves financial transaction from the data source

}

getExpenseTransaction(){

//retrieves expense transaction from the data source

}

persist(){

//saves to the data source

}

}

Public class DomainController(){

getPaymentTypes(){

//works on backend to run the query to get payment types

}

addIncomeTransaction(){

//works on backend to add a new Income Transaction object

}

addExpenseTransaction(){

//works on backend to add a new Expense Transaction object

}

saveIncomeTransaction(){

//works on the backend to save the object created by Income Transaction data

}

saveExpenseTransaction(){

//works on the backend to save the object created by Expense Transaction data

}

IncomeTransID(){

// self message from domain controller to domain controller

}

IncomeTransHST(){

// self message from domain controller to domain controller

}

ExpenseTransID(){

// self message from domain controller to domain controller

}

ExpenseTransHST(){

// self message from domain controller to domain controller

}

TotalIncome(){

// self message from domain controller to domain controller

}

TotalHSTCollected(){

// self message from domain controller to domain controller

}

TotalExpense(){

// self message from domain controller to domain controller

}

TotalHSTPaid(){

// self message from domain controller to domain controller

}

Total(){

// self message from domain controller to domain controller

}

HSTOwed(){

// self message from domain controller to domain controller

}

}

Public class UIController(){

incometransaction(){

//displays user the data

}

addIncomeTransaction(){

//prompts the user to add Income Transaction

}

addExpenseTransaction(){

//prompts the user to add Expense Transaction

}

enterIncomeTransaction(){

//prompts the user to enter IncomeTransactiondata

}

saveIncomeTransaction(){

//prompts the user to save IncomeTransactiondata

}

}

Public class PaymentType(){

int PaymentTypeID;

string PaymentTypeDescription;

IncomeTransaction IncTrans()

getPaymentTypes(){

// works on backend to run the query to get Payment Types

}

create(){

// creates a new object of this class

}

}

Public class ExpenseTrasaction(){

int ExpenseTransID;

string ExpenseTransDescription;

double ExpenseTransAmount;

int ExpenseTransDate;

double ExpenseTransHST;

}

Public class IncomeTransaction(){

int IncomeTransID;

string IncomeTransDescription;

double IncomeTransAmount;

int IncomeTransDate;

double IncomeTransHST;

PaymentType paytype;

}