**Money Module**

Sara and Ted’s bank account looks good but at a glance, they can’t easily see which transactions are related to their online business and which are from kiosk.

They find that at the end of the month manually. They are currently using excel to record details from their bank accounts and credit card to figure it out. Sara and Ted would like to record a description of the transaction indicating whether or not the transaction was income or expense and more details of what the transaction was for. They usually pay for supplies using their credit card. Their customers pay by cash, credit card or debit card. Online sales are paid by credit card.

Once Sarah and Ted’s business grows to $30,000 in sales, they will be required to collect and pay HST. For this reason, Sarah and Ted would like to know how much they are paid and collected in HST.

They would like a more robust system than Excel to manage these details.

Your task is to read the User Story below and create a class diagram and sequence diagrams to support the scenario.

**User Story**

As the owner of this business, I would like to record my financial transactions so that I can quickly see how much money I’ve made and what I owe the government in HST. For my purposes, my expenses subtracted from my income.

Acceptance Criteria:

1. Must be able to query transactions by date and provide a total profit and total HST
2. Must be able to query transactions by date and by source of the income.
3. Differentiate between expenses and income – provide a running total of income – expenses
4. Calculate the HST owed to the government by subtracting HST paid on expenses from HST collected on income.

A screenshot of a computer screen

Description automatically generatedb

**Use Case Descriptions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Create an income transaction** | | |
| Triggering Event | Money coming into the business. | | |
| Brief Description | Allows the Owner to record an income transaction. | | |
| Actors | Owner | | |
| Related Use Cases |  | | |
| Preconditions | Owner has opened the Main Menu. | | |
| Post Conditions | Income transactions are saved and added to financial totals. | | |
| Flow of activities | Actor | | System |
|  |  | Requests to add a new income transaction | Displays a calendar.  Prompts for description, transaction amount and HST amount.  Displays a list of income sources and prompts for selection. |
|  |  | Select the transaction Date.  Selects the income source.  Enter the transaction description.  Enters the transaction amount. | Verifies that date is selected.  Verifies that description and amount have been entered.  Verifies that the income source was selected.  Calculates HST.  Displays income transaction.  Prompts to save. |
|  |  | Request to save | Saves the transaction and returns to the main menu. |
| Exception Conditions | * Owner chooses to cancel adding the transaction | | |

A diagram of a project

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Create an expense transaction** | | |
| Triggering Event | Money leaving the business. | | |
| Brief Description | Allows the Owner to record an expense transaction. | | |
| Actors | Owner | | |
| Related Use Cases |  | | |
| Preconditions | Owner has opened the Main Menu. | | |
| Post Conditions | Expense transaction is saved and added to financial totals. | | |
| Flow of activities | Actor | | System |
|  |  | Requests to add a new expense transaction | Displays a calendar.  Prompts for description, transaction amount and HST amount. |
|  |  | Selects the transaction Date  Enters the transaction description  Enters the transaction amount | Verifies that date is selected.  Verifies that description and amount have been entered.  Calculates HST.  Displays financial transaction.  Prompts to save. |
|  |  | Request to save | Saves the transaction and returns to the main menu |
| Exception Conditions | * Owner chooses to cancel adding the transaction | | |

A diagram of a diagram

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Name** | **Query Financial Transaction** | | | |
| Triggering Event | Owner requires a list of transactions for a date period including Total Income, Total Expenses, Total Profit, Total HST collected, Total HST Paid, HST Owed | | | |
| Brief Description | Allows the Owner to retrieve financial totals for a specified date range | | | |
| Actors | Owner | | | |
| Related Use Cases |  | | | |
| Preconditions | Owner has opened the Main Menu. | | | |
| Post Conditions | Financial transactions are retrieved, totaled and displayed | | | |
| Flow of activities | Actor | | System | |
|  | 1. | Requests to query financial transactions | | Displays a calendar. |
|  | 2. | Selects date range. | | Verifies that dates are selected.  Retrieves transactions in the specified date range.  Calculates and displays total income, total expenses, Profit, total HST Owed.  Prompts to exit |
|  | 3. | Request to exit | | returns to the main menu |
| Exception Conditions |  | | | |

A diagram of a project

Description automatically generated

.h prototypes

.h files

IncomeTransaction.h

public class IncomeTransaction {

int incTransID;

IncomeSource incomeSource;

int incTransdate;

String incTransDesc;

float incTransAmount;

getIncomeTransaction()

}

ExpenseTransaction.h

public class ExpenseTransaction {

int expTransID;

int expTransDate;

String exptransDesc;

float expTransAmount;

getExpenseTransaction()

}

IncomeSource.h

public class IncomeSource {

int incSrcID;

String incSrcDesc;

getIncomeSources()

}

UIController.h

public class UIController {

addIncomeSource() {

// sends request to create an income transaction

// returns an income source list

}

enterIncomeTransaction(transDate,incSrc,transAmount,transDesc) {

// sends request to add an income transaction

// returns income transaction

}

confirm(){

// sends request save transaction

}

createNewExpenseTransaction() {

// sends request to create an expense transaction

}

enterExpenseTransaction(expTransDate,expTransAmount,expTransDesc) {

// sends request to add an expense transaction

// returns expense transaction

}

saveTransaction()

retrieveFinancialTrans() {

// displays a calendar

}

getFinancialTrans(startDate,endDate) {

// sends request to query financial transactions

// displays financial transactions

}

exit() {

// exits

}

}

DomainController.h

public class DomainController.h {

getIncomeSource() {

// sends request to retrieve income sources

// returns a list of income sources

}

createIncomeTransaction(transDate,incSrc,transAmount,transDesc) {

// sends request to create income transaction

// returns income transaction

}

saveIncomeTransaction(){

// sends request save transaction

}

createExpenseTrans(expTransDate,expTransAmount,expTransDesc) {

// sends request to create expense transaction

// returns expense transaction

}

generateExpTransID()

calculateHST()

getFinancialTrans(startDate,endDate) {

// sends request to query financial transactions

// returns financial transactions and totals

}

calculateHST(){

// calculates HST

}

calculateIncTotal(){

// calculates income total

}

calculateHSTPaid(){

// calculates HST paid

}

calculateExpTotal(){

// calculates expense total

}

calculateHSTOwed(){

// calculates HST Owed

}

calculateProfit(){

// calculates profit

}

}

EntityManager.h

public class EntityManager {

persist(object)

// saves

get(class, selection criteria)

// retrieves from the table

}