Department of Information Technology (Chai Ward)

IT524122 HD in Cybersecurity

ced Scripting Technology

ssignment

Assignment Due Date

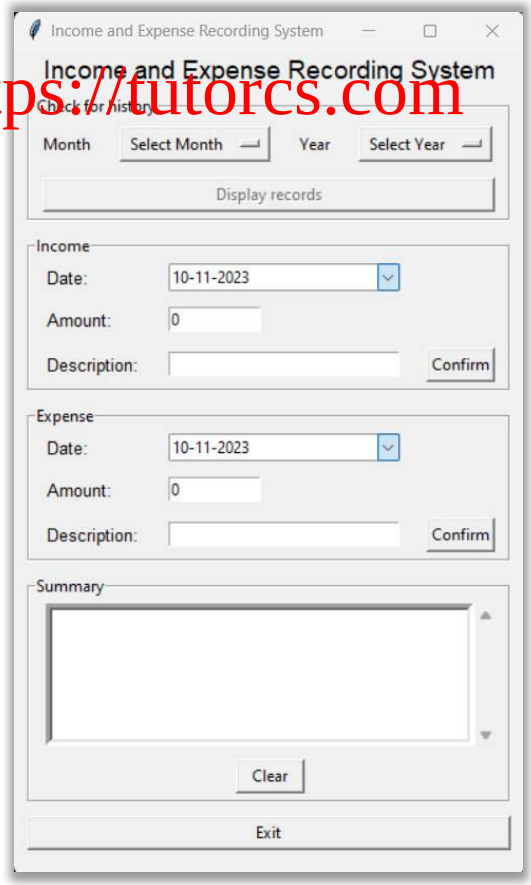
11 Dec 2023 Monday e

Objective

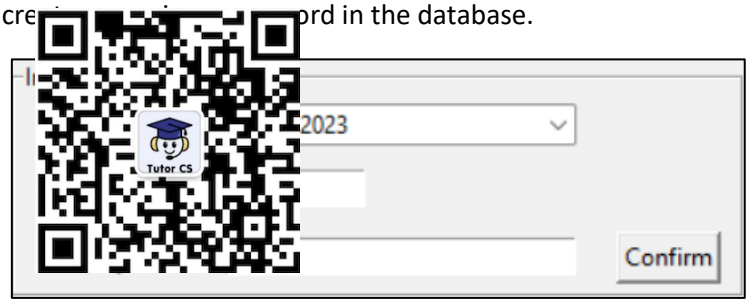
This assignment is implementing an **Income and Expense Recording System** using Object Oriented Programming Techniques. Your coding must be able to show concepts such as Abstraction, Encapsulation, Inheritance, Polymorphism, and Database Connection. Your program must be able to do the following:

1. Establish connection to MySQL database server and perform read/write function.
2. Display income and expense history.
3. Create income and expense records.

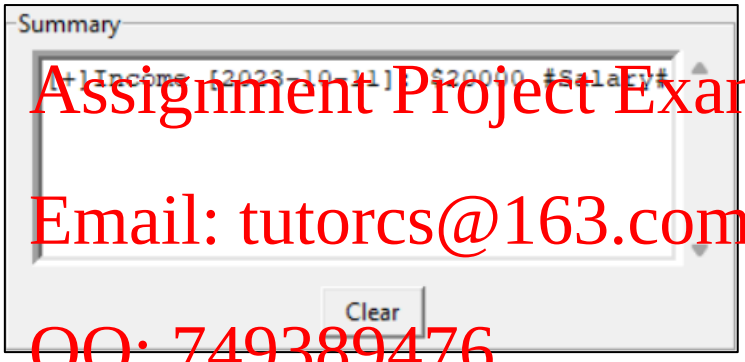
Assignment requirements



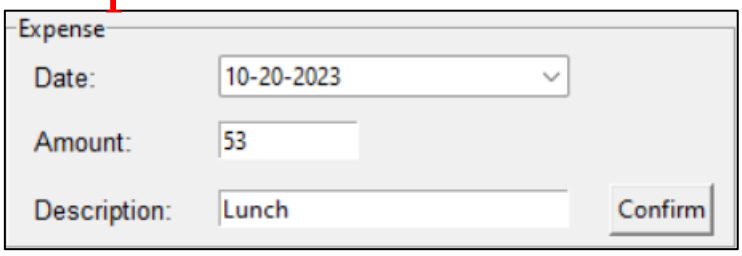
Add Income Record: The user can add a new income record by choosing a date or using the default date value, filling in the amount and description in the Income label frame, and then clicking the **Confirm** button to cre ord in the database.



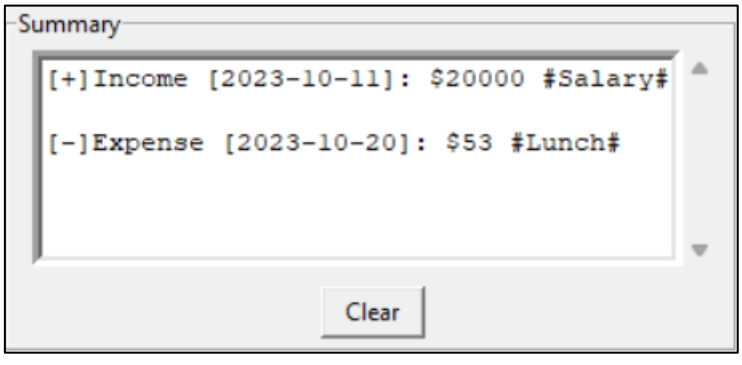
The new income record would be displayed in the Summary scrolled textbox, and a new record would be created in the database.



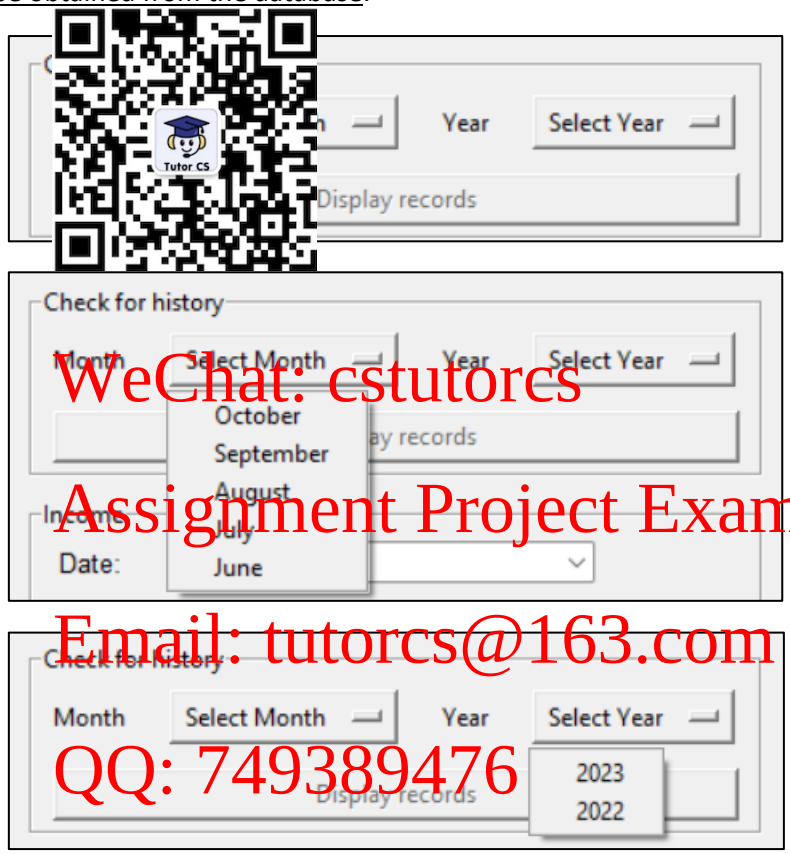
Add Expense Record: The user can add a new expense record by choosing a date or using the default date value, filling in the amount and description in the Expense label frame, and then clicking the **Confirm** button to create a new expense record in the database.



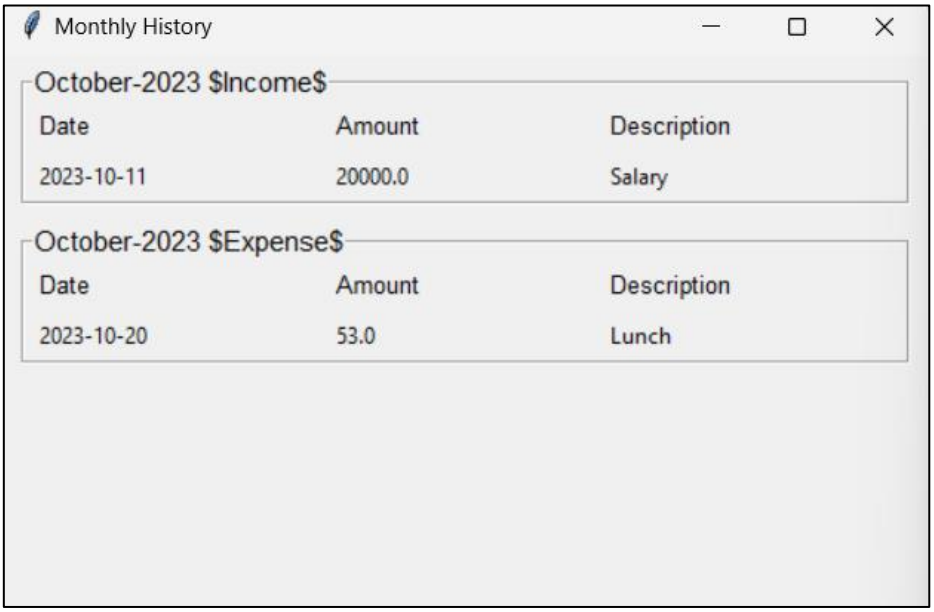
The new expense record would be displayed in the Summary scrolled textbox, and a new record would be created in the database.



Check Monthly History: By choosing both Month and Year, the "Display records" button will be released. The content of both "Select Month" and "Select Year" will be modified whenever a new record is created, the Month option is selected, or the Year option is selected. The value of Month and Year should be obtained from the database.

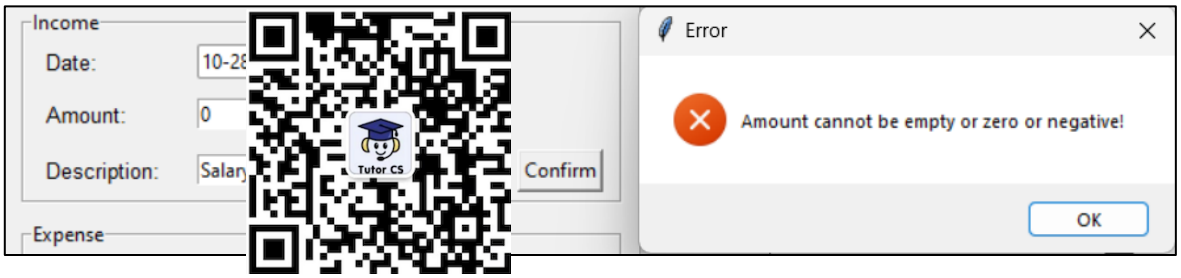


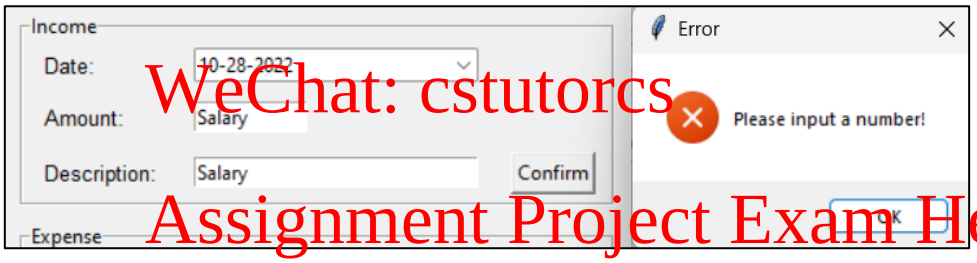
Click the "Display records" button to check the record for the selected month and year. The record of a specific month-year should be retrieved from the database and displayed properly as shown.



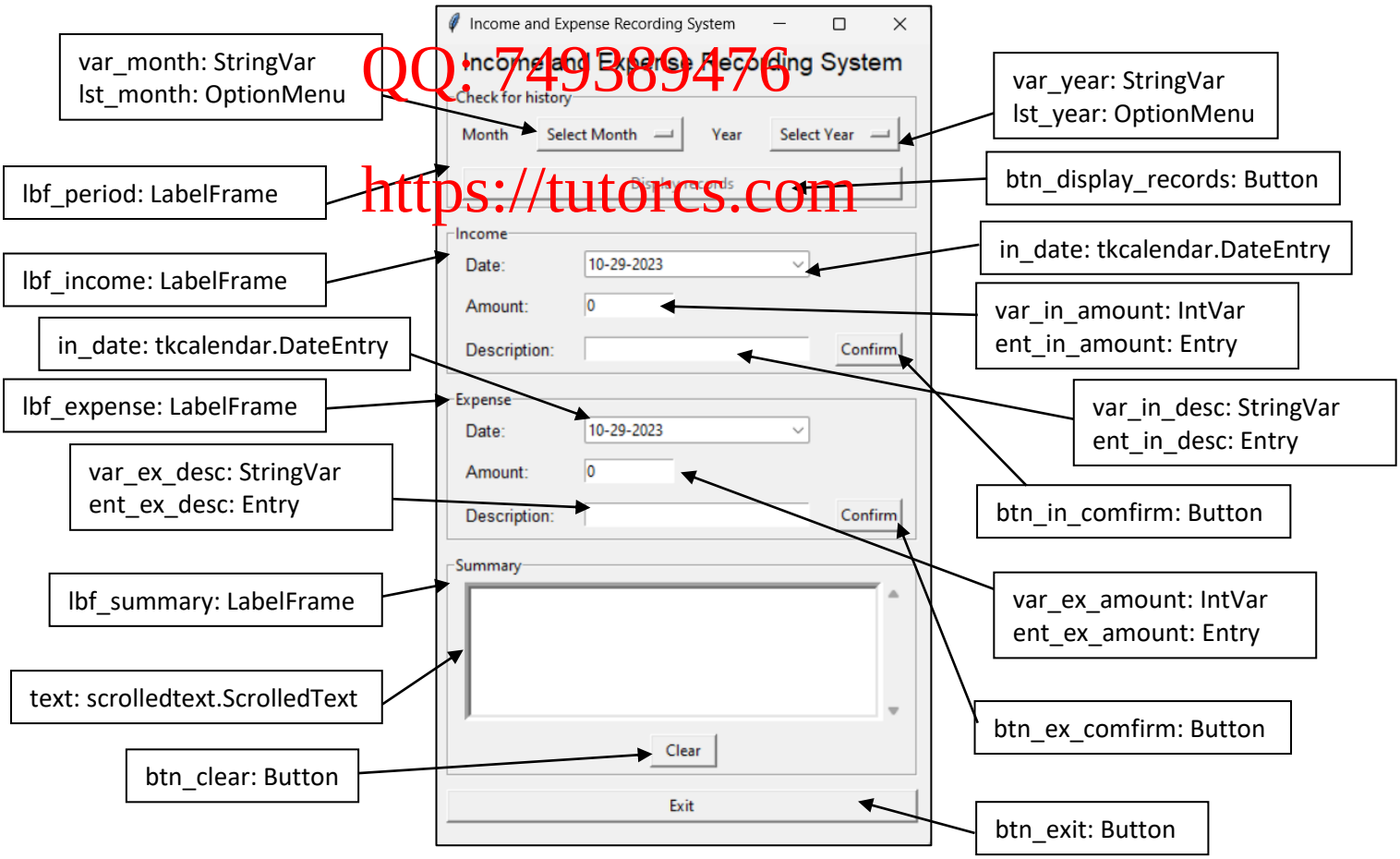
Clear button: All the input records, summary, and selected history options will be cleared.

Error Checking: The system should provide input error checking in the following cases:
For income and expense records creation, the amount value should be inputted positive, and a non-zero value. If not, the following error message will be displayed.





Programming Requirements of the Assignment



Layout for *Application*

You are given the above ***UI layout*** and ***a partially developed application class*** inherited from the Tkinter class for the above requirements. The class diagram of the required library is shown in the next section. You must complete both the application and the class library so that your application produces the same result as the above requirement.

File structure and des

Files	**Cla**	
application.py	Dis Inc rdSystem	• A partially developed application that contains the main GUI implemented in the class **IncomeAndExpenseRecordSystem(tkinter.Tk)**. You must study the code and complete the rest of the program logic. • You need to add another class (DisplayHistory) in this file to design UI related to displaying monthly history records.
db.py	ConnectDB	• File used to perform database connection and operations. • This file contains all methods to retrieve data from the database and inert record to the database
library.py	Record Income_Record Expense_Record	• Supplier classes (i.e. **Record, Income_Record, Expense_Record**) should be implemented in this file

Things to do

This assignment is divided into two parts: Database and Tkinter GUI.

Study the provided code in each python file and all the comments.

TODO comments are ...tification of the parts that need to be completed by you.

Part 1 – Database and

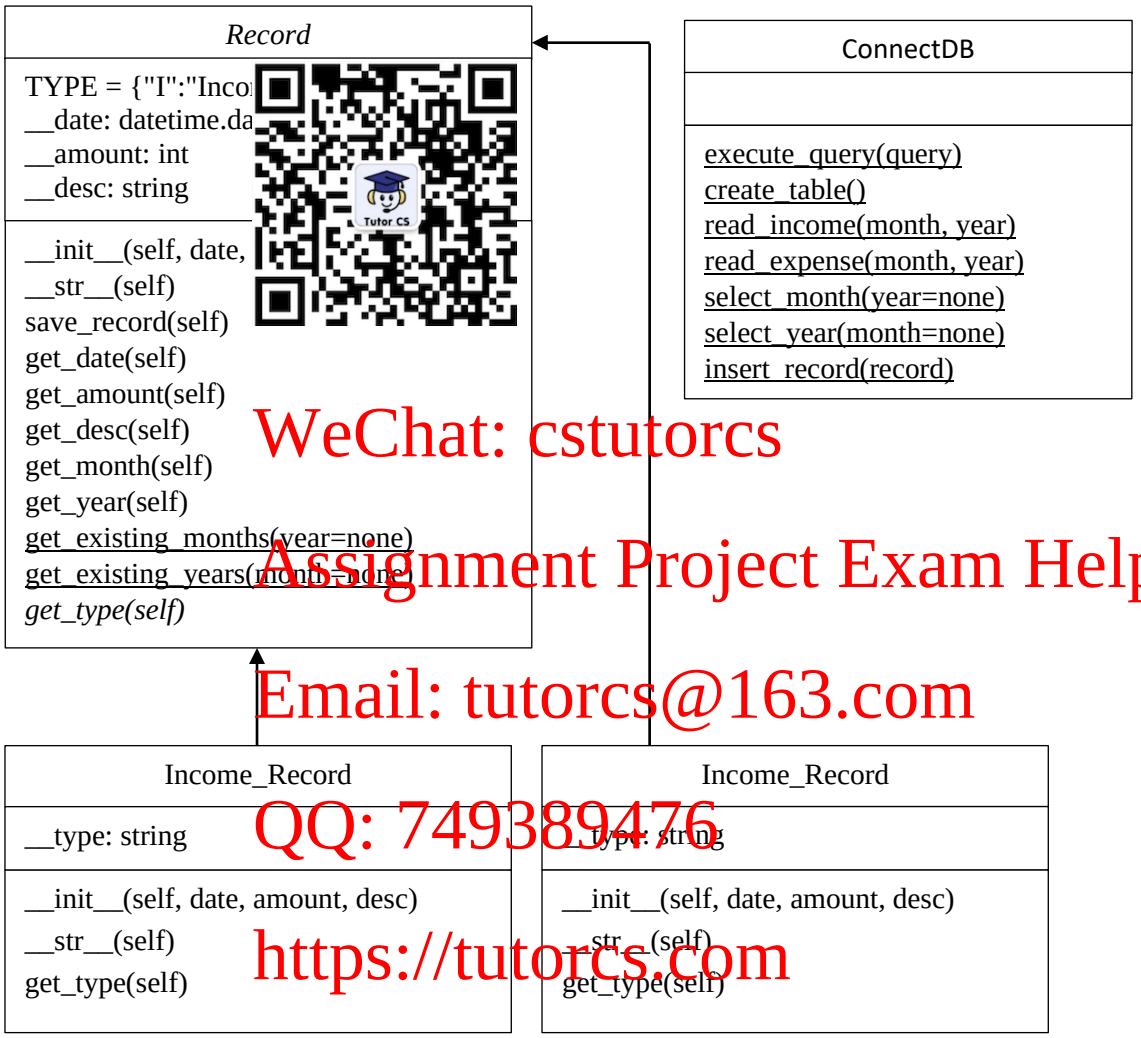
(a) Complete the ... file **db.py**. Refer to *Database Design* for the necessary information.

(b) Completed th... the file **library.py**.

Part 2 – Tkinter GUI

(a) Complete the code segments of **DisplayHistory** class and **IncomeAndExpenseRecordSystem** class.

Description of Class Library



(a) Create an abstract class **Record** that has:

Attributes:

- **__date** a non-public string to store the date of an income or expense record
- **__amount** a non-public string to store the income or expense amount of a record
- **__desc** a non-public dictionary to store the remarks of a record
- **TYPE** a dictionary to store the strings that distinguish between different record types

Methods:

- **__init__** initialize the **date**, **amount,** and **desc** attributes. call save_record to save a record in the database when a Record object is created
- **__str__** returns the output string of a record as shown in the sample output (e.g., [2023-10-20]: $20 #Lunch#)
- **save_record** a method to insert a new record to the database by calling the method in db.py
- **get_date** returns date for a record
- **get_amount** returns amount for a record
- **get_desc** returns description for a record

- **get_month** return the month name (e.g., October) using the **date** value
- **get_year** return the year (e.g., 2023) using the **date** value
- **get_existing_months** a static method to retrieve a list of months with existing records in the database. If the month variable is received, include the he where clause in the select query.
- **get_existing_** ethod to retrieve a list of years with existing records abase. If the month variable is received, include the he where clause in the select query.
- **get_type** t method used to force all child classes to implement od. Since the type cannot be determined without ils given.

(b) Create a class **Income_Record** inherits from class **Record** that has:

Attributes:

- **__type** a non-public string to store the type of record

Methods:

- **__init__** initialize the **type** attribute
  super method call is used to call the **parent class __init__ method** to initialize the name attributes.
- **__str__** return the output string of a record by adding the record type heading
  super method call is used to call the **parent class __str__ method** to output the record details.
- **get_type** Return type for a record

(c) Create a class **Expense_Record** inherits from class **Record** that has:

Attributes:

- **__type** a non-public string to store the type of record

Methods:

- **__init__** initialize the **type** attribute
  super method call is used to call the **parent class __init__ method** to initialize the name attributes.
- **__str__** return the output string of a record by adding the record type heading.
  super method calls to call the **parent class __str__ method** to output the record details.
- **get_type** Return type for a record

(d) Create a class **ConnectDB** that has:

Methods:

- **execute_query** establish database connection using mysql.connector library to connect database "in_ex_record_system"
  create a cursor and use the cursor to execute the received query variable and return the fetched result
- **create_table** create a table "record" with the required column (refer to *Database Design* section)
- **read_income** return income records of specific month and year from the database
- **read_expense** return expense records of specific month and year from the database

- **select_month** return a list of months that have existing records in the database
- **select_year** return a list of years that have existing records in the database
- **insert_record** Insert a new income or expense record into database, providing all detailed information

Apart from the above, you must complete the methods in the Tkinter-related classes, **DisplayHistory** and **InputRecordSystem** in application.py.

Instructions to students

1. This is an **End-of-Module Assessment** and the weighting of this assignment is **20%** of the Module Mark.
2. This assignment should be done by each student **individually**. Plagiarism will be treated seriously. All assignments that have been found involved wholly or partly in plagiarism (no matter these assignments are from the original authors or from the plagiarists) will score **ZERO** mark.
3. You must use Python 3.9 or above the develop the program.
4. Your source code must follow the coding standard stated in PEP 8 – Style Guide for Python Code. Marks may be deducted if the coding standard is not followed.
5. You are required to hand in the following:
   5.1 A test plan showing the **evidence of testing**.
   5.2 Source code for the entire program, with adequate comments.

ID	Test Case Name	Procedure	Expected Output	Result
1	Add income record	1. In the income label frame, input the date, amount, and description 2. Click “Confirm” button	A new record should be added to the database. The corresponding output message is displayed in the summary section. The month and year option in check for the history label frame is updated accordingly	Pass / ~~Fail~~
	…			
5	Input an empty amount	1. In the income label frame, leave the amount blank 2. Click “Confirm” button	An error message box “Amount cannot be empty or zero or negative!” should be prompted out.	Pass / ~~Fail~~

6. Submit all your works in a single **ZIP** file under the name of your student ID and name (e.g., 22XXXXXXX-ChanDaiMan.zip) to Moodle before the assignment deadline. Marks will be deducted for late submissions (e.g., during program demonstration) or even score **ZERO** mark.

Mark Distribution

- System Implementation (75%)

File	Implementation	Marks Distribution
	Database and supplier classes	
db.py		25%
library.py	class Record class Income_Record class Expense_Record	15%
	Part 2 – Tkinter GUI	
application.py	class DisplayHistory class IncomeAndExpenseRecordSystem	60%

- Programming Style (10%)
- Test Plan (15%)

Database Design

Table – **record**

Column	Type	PK	Modifiers
id	INT	✓	NOT NULL AUTO_INCREMENT
rcMonth	VARCHAR(10)		NOT NULL
rcYear	VARCHAR(10)		NOT NULL
reDate	VARCHAR(20)		NOT NULL
amount	DOUBLE		NOT NULL
description	VARCHAR(100)		NOT NULL
record_type	VARCHAR(7)		NOT NULL